
HUEBER SOFTWARE TASCHENBUCH
SPIELE
FÜR IHREN
ZX 81



von Mark Charlton

Herausgeber der Reihe
Tim Hartnell

Max Hueber Verlag

Titel der englischen Originalausgabe:
Games for your ZX-81
Übersetzung: Benjamin Hembus
Umschlagfoto: Fred-Jürgen Rogner, London
Umschlagentwurf: Agentur Cooperation, München

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Charlton, Mark:
Spiele für Ihren Sinclair ZX 81 / Mark Charlton
[Übers.: Benjamin Hembus]. – 1. Aufl. – München
[i.e. Ismaning]: Hueber, 1984.
(Hueber-Software: Taschenbuch)
Einheitssacht.: Games for your ZX-81 «dt.»
ISBN 3-19-008212-X

1. Auflage 3 2 1

Satz: Typo Dreitausend, München
Druck: RMO-Druck, München · Printed in Germany
ISBN 3-19-008212-X

Inhalt

Tim Hartnell – Herausgeber der Reihe

Tim Hartnell ist ein führender Computer-Experte und einer der Großen unter den bekannten Namen der populären Elektronik-Publizistik. Zu seinen bekanntesten Büchern gehören „49 explosive Spiele für den Sinclair ZX 81“, „Entdecken Sie die unendlichen Dimensionen Ihres ZX 81“ und „Spectrum ohne Grenzen“ (sämtlich erschienen bei Hueber Software).

Mark Charlton – der Verfasser

Mark Charlton hat unter anderem "The Gateway Guide to the ZX 80 und ZX 81" geschrieben. Ein großes Leserpublikum hat er als Kolumnist der Zeitschrift "Interface".

Dank

Sehr herzlich möchte sich der Autor bei den Kollegen bedanken, die Beiträge für dieses Buch beigesteuert haben: Paul Toland, Dilwyn Jones, Nick Wilson, Neil Streeter, Graham Charlton, Tim Rogers, Chris Callender, Clive Gifford, Alistair Miller und Jim Archer.

Vorwort des Herausgebers

Ihr Computer wartet darauf, Sie auf die Probe zu stellen. Spiele mit bewegter Grafik, Denkspiele, Ratespiele und Rätsel – alles zu Ihrer Unterhaltung.

Der Band enthält eine Vielfalt von Spielen. Die Programme sind verfaßt von einigen der begabtesten Jungprogrammierer bei uns und bieten eine ganze Reihe von Methoden, Programmierprobleme zu lösen.

Wenn Sie sich mit den Listings näher befassen, lernen Sie viele neue Kniffe und Wege kennen, die Sie für Ihre eigene Programmierung nutzen können. Und sobald Sie die Programme in der vorliegenden Form beherrschen, wollen Sie vielleicht Verbesserungen ausprobieren. Ein „perfektes" Programm gibt es bekanntlich nicht, also können die gebotenen Programme durch Ihre Talente nur gewinnen.

Sie brauchen jetzt nur noch umzublättern und die Programme einzugeben. Ich kann nur hoffen, daß Ihnen das Spielen soviel Spaß macht wie vorher uns, den Band zusammenzustellen.

Tim Hartnell, Herausgeber der Reihe
London
März 1983



Stadt

STADT ist Paul Tolands Version des Spiels, in dem ein Flugzeug eine Stadt anfliegt und die Wolkenkratzer wegbomben muß, um landen zu können. So, wie es im Folgenden gelistet ist, hat das Programm einen sehr hohen Schwierigkeitsgrad (um die Wahrheit zu sagen: es ist kaum zu schaffen). Eine Spielerleichterung können Sie sich verschaffen, indem Sie die Wolkenkratzer etwas niedriger halten. Dies geschieht, indem Sie aus der 12 in Zeile 30 eine niedrigere Ziffer machen, eine 8 zum Beispiel. Wenn Sie lieber ein Dorf als eine Stadt anfliegen wollen, versuchen Sie es mit 10 FOR 1=8 TO 25.
Die Stadt wird nach Belieben immer wieder ganz anders aufgebaut. Zum Bomben-Abwurf drücken Sie irgend einen Knopf, nur nicht BREAK. Wenn Sie eine Bruchlandung bauen, drücken Sie irgend einen Knopf außer N; dann kann das Spiel von neuem beginnen. Wollen Sie das Programm im LOADED-Status als selbststartendes Spiel benutzen, so speichern Sie es mit GOTO 340.

```
   1 REM "STADT" PAUL TOLAND
   5 RAND
  10 FOR I=1 TO 31
  20 LET C$=CHR$ (RND*10+128)
  30 FOR J=21 TO 21-INT (RND*12)
 STEP -1
  40 PRINT AT J,I;C$
  50 NEXT J
  60 NEXT I
  70 LET B=0
  90 LET PL=1
  95 LET P=PEEK 16396+256*PEEK 1
6397+1
```

```
  97 LET PL=P+1
 100 POKE PL,0
 110 LET N=PEEK (PL+1)
 120 IF N=0 THEN GOTO 150
 130 IF N<>118 THEN GOTO 250
 140 LET PL=PL+1
 150 LET PL=PL+1
 160 POKE PL,130
 170 IF INKEY$>"" AND B=0 THEN L
ET B=PL
 180 IF B=0 THEN GOTO 240
 190 POKE B,0
 200 LET B=B+33
 210 LET D=B
 220 IF PEEK B>0 OR B>P+724 THEN
 LET B=0
 225 POKE D,28
 227 LET I=1
 230 POKE D,0
 240 GOTO 100
 250 POKE PL,131
 251 POKE PL,132
 252 LET I=1**1
 253 POKE PL,133
 254 LET I=1**1
 255 POKE PL,134
 256 LET I=1**1
 257 POKE PL,135
 258 LET I=1**1
 259 POKE PL,23
 260 PRINT AT 8,10;"BRUCHLANDUNG
"
 270 GOTO 290
 280 PRINT AT 8,5;"SIE HABEN""S
HEIL GESCHAFFT  "
 290 PRINT TAB 5;"NOCH EINMAL? (
J/N)"
 300 IF INKEY$="" THEN GOTO 300
 310 IF INKEY$="N" THEN STOP
 320 CLS
 330 RUN
 340 SAVE "STADT"
 350 RUN
```

Falle

Ein raffinierter Fallensteller ist Dilwyn Jones, der dieses 1K Graphics-Spiel entwickelt hat. Sie sind das arme Opfer, verkörpert durch das +-Zeichen, das sich, geführt von den "5"- (Links), "6"- (Abwärts), "7"- (Aufwärts) und "8"- (Rechts) Tasten, über den Schirm bewegt. Für jede einzelne Bewegung werden Sie mit einem Punkt belohnt. Der Computer versucht, Sie in die Falle zu locken, indem er Sie mit seinen Steinen einkreist. Sie haben das Spiel verloren, wenn Ihnen kein Freiraum mehr für eine Bewegung bleibt. (Gewinnen ist, wie im wirklichen Leben, unmöglich.) Ihr Punktestand wird am Schluß angezeigt.

```
   10 RAND
   20 LET A=0
   30 PRINT AT 14,0;"████████████
█";AT 2,0;"█████████████"
   32 REM 2 MAL 13*GRAPHIC SPACE
   40 FOR X=0 TO 10
   50 PRINT "█";TAB 12;"█"
   60 NEXT X
   70 LET X=INT (RND*11)+1
   80 LET Y=INT (RND*11)+3
   90 PRINT AT Y,X;
  100 IF PEEK (PEEK 16398+256*PEE
K 16399)=128 THEN GOTO 190
  110 PRINT "+";AT Y+(RND<.5)-(RN
D<.5),X+(RND<.5)-(RND<.5);"█" AN
D PEEK (PEEK 16398+256*PEEK 1639
9)<>21
  120 LET A=A+1
  130 LET A$=INKEY$
  140 IF A$<"5" OR A$>"8" THEN GO
TO 130
  150 PRINT AT Y,X;" "
  160 LET X=X+(A$="8")-(A$="5")
```

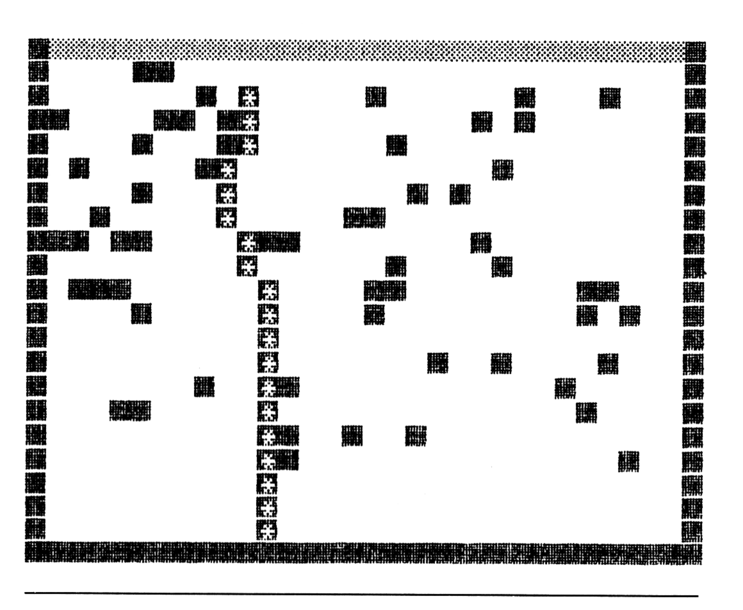
```
170 LET Y=Y+(A$="6")-(A$="7")
180 GOTO 90
190 PRINT A
195 STOP
200 SAVE "TRA[illegible]"
210 GOTO 1
```

Panzerangriff

In diesem Spiel, das der alte Panzerhase Nick Wilson für den 16K ZX81 entwickelt hat, müssen Sie vom unteren zum oberen Bildschirm-Rand gelangen. Das ist ein risikoreiches Unternehmen, da der Weg mit Panzerminen gepflastert ist, verkörpert durch schwarze Vierecke. Wenn Sie das Spiel mit RUN starten, erscheint auf dem Schirm der Angriffskurs, gesäumt von einem Explosiv-Gürtel.
Der Panzer, dargestellt durch einen inversen Stern, bewegt sich vom Start des Spiels an aufwärts, von Ihnen gesteuert durch die "O"-Taste (Rechts) und die "1"-Taste. Über das Schlachtfeld sind 60 Minen verteilt. Sie können das Spiel variieren, indem Sie die 65 in Zeile 75 nach Ihrem Belieben ändern; je niedriger die Zahl, umso einfacher das Spiel. Mit "Y" beginnen Sie ein neues Spiel.
Wenn Sie zum glorreichen Ende der Aktion den oberen Bildschirm-Rand erreichen, erscheint Ihre „Panzer-Kommando"-Bewertung. Das Ganze ist ein temporeiches Spiel. Wenn Sie es zum Eingewöhnen verlangsamen wollen, müssen Sie 107 LET G = SIN PI hinzufügen. Das Graphik-Zeichen in Zeile 30 erzeugen Sie durch die "H"-Taste.
Die Zeilen 12 bis 85 geben den Kurs und den Explosiv-Gürtel an; 100 bis 170 bewegen den Panzer nach oben und nach Links oder Rechts; 150 und 160 lesen die Konsulen und bewegen den Panzer; 200 bis 230 lösen eine Explosion am Panzer aus; und die Zeilen 245 bis 360 beenden das Spiel und drucken Ihr Ergebnis aus.

```
  10 REM PANZERANGRIFF
  11 REM NICK WILSON
  12 RAND
  13 FAST
  14 LET S=0
  15 CLS
  20 FOR I=0 TO 31
  30 PRINT AT 0,I;"▒";AT 21,I;"█
"
  40 NEXT I
  50 FOR I=0 TO 20
  60 PRINT AT I,0;"█";TAB 31;"█"
  70 NEXT I
  75 FOR I=1 TO 65
  80 PRINT AT INT (RND*17)+1,INT
 (RND*29);"█"
  85 NEXT I
  86 LET M=20
  90 LET X=INT (RND*28)+1
  95 SLOW
 100 PRINT AT M,X;
 105 LET S=S+230
 110 IF PEEK (PEEK 16398+256*PEE
K 16399)=128 THEN GOTO 200
 120 IF M=0 THEN GOTO 300
 130 PRINT "▣"
 140 LET M=M-1
 150 IF INKEY$="1" AND X>0 THEN
LET X=X-1
 160 IF INKEY$="0" AND X<30 THEN
 LET X=X+1
 170 GOTO 100
 200 FOR I=1 TO 10
 210 PRINT AT M,X;"*"
 215 PRINT AT M,X;"▣"
 220 PRINT AT M,X;" "
 230 NEXT I
 240 CLS
 245 LET S=S-300
 250 PRINT AT 10,0;"PANZER EXPLO
DIERT - PUNKTE:";S;AT 15,3;"NOCH
 EIN SPIEL?"
```

```
 260 GOTO 340
 300 FOR I=1 TO 23
 310 SCROLL
 320 NEXT I
 330 PRINT AT 10,0;"GUT GEMACHT
- SIE HABEN ";S;AT 11,0;"PUNKTE
ERZIELT.";AT 13,0;"NOCH EIN SPIE
L?"
 340 IF INKEY$="" THEN GOTO 340
 350 IF INKEY$<>"N" THEN RUN
 360 STOP
 370 SAVE "TANK"
 380 RUN
```



20.000 Meilen unter dem Meer

Alle ZX 81-Freunde kennen das Problem, daß bei den in BASIC geschriebenen Programmen die graphischen Elemente einen ziemlich geruhsamen Bewegungs-Rhythmus haben; das hat zur Folge, daß Graphics-Spiele schwer zu programmieren sind. "Space Invaders" zum Beispiel verliert viel von seinem Reiz, wenn sich die Raumschiffe nur alle 10 Sekunden bewegen.
Wollen Sie nun trotzdem bei BASIC bleiben und nicht auf den Maschinen-Code ausweichen, so ist der einzige Weg, die graphischen Elemente zu einem realistischen Bewegungs-Rhythmus zu bringen, die Programme so kurz wie möglich zu halten und so wenig Elemente wie möglich auf dem Schirm zu bewegen. Genau das hat Neil Streeter mit diesem Programm geschafft. Die einzigen Dinge, die sich hier bewegen, sind die Taucherglocke und der Octopus, und das eine wie der andere wird von je einem einzigen Element verkörpert. Dadurch wird der ganze Ablauf enorm beschleunigt.
Der Bewegungsablauf liegt zwischen den Zeilen 170 und 290 und ist in seinen Elementen auf das Notwendigste reduziert: Kontrolle der gedrückten Taste, Kontrolle der Schatzkiste, Kontrolle des Octopus. Nicht nur das Ergebnis, sondern auch alle anderen Mitteilungen werden ausgespart und an das Spielende versetzt. Wenn Sie nun versuchen, alle Bewegungsabläufe so kurz wie möglich und so nahe wie möglich an der Ausgangsposition zu halten, werden Sie auf ein sehr befriedigendes Spiel-

Tempo kommen. Die Elemente in Zeile 8Ø sind Graphics 8, Graphics 3, drei inverse "", Graphics 4, elf-, Graphics 2, Inverse Space, zwei inverse "", A, drei Inverse "" und 13 – bis Schluß.

```
  10 PRINT "20 000 MEILEN UNTER DEM MEER      "
  20 PRINT "                              KOENNEN SIE IN IHRER TAUCHER-   GLOCKE DIE 3 SCHATZTRUHEN BERGENOHNE VOM RIESIGEN KRAKEN GEFAN- GEN ZU WERDEN?"
  30 PRINT ,,"SCHWIERIGKEITSGRAD (1-4):","1=SCHWER 2+3=MITTEL 4=LEICHT"
  32 INPUT LEOD
  33 IF INT LEOD<>LEOD OR LEOD>4 THEN GOTO 32
  35 LET LEOD=(LEOD+1)/2
  40 PRINT ,,"MIT DEN TASTEN 5-8 STEUERN SIE  DIE TAUCHERGLOCKE",,,,,"MIT TASTE 6 BEGINNT DAS SPIEL."
  50 PRINT "                              "
  60 IF INKEY$="" THEN GOTO 60
  70 CLS
  74 LET COUNT=0
  75 LET TEST=0
  80 PRINT TAB 15;" ";TAB 13;"     ","-----------    A   -------------"
  85 LET D$="A"
  90 LET X=15
 100 LET Y=2
 110 PRINT AT 21,0;"                                "
 120 PRINT AT 20,3;" ";TAB 7;" ";TAB 25;" "
 130 LET OX=INT (RND*32)
```

```
 140 LET OY=INT (RND*10+8)
 150 PRINT AT OY,OX;"O"
 160 LET TS=0
 165 LET SC=0
 170 LET X1=X
 180 LET Y1=Y
 190 LET X=X+(INKEY$="8" AND X<3
1)-(INKEY$="5" AND X>0)
 200 LET Y=Y+(INKEY$="6" AND Y<2
0)-(INKEY$="7" AND Y>2)
 205 LET COUNT=COUNT+1
 210 PRINT AT Y1,X1;" ";AT Y,X;
 212 IF Y<>2 THEN LET TEST=1
 215 IF X=15 AND Y=2 AND TEST<>0
 THEN GOSUB 1500
 220 LET CH=PEEK (PEEK 16398+256
*PEEK 16399)
 225 IF CH THEN GOSUB 1000
 226 PRINT D$
 230 LET OX1=OX
 240 LET OY1=OY
 250 LET OX=OX+(X>OX)/LEOD-(X<OX
)/LEOD
 260 LET OY=OY+(Y>OY)/LEOD-(Y<OY
)/LEOD
 262 IF INT OX=X AND INT OY=Y TH
EN GOTO 3000
 265 IF OY>20 THEN LET OY=20
 266 IF OY<3 THEN LET OY=3
 270 PRINT AT OY1,OX1;" ";AT OY,
OX;"O"
 290 GOTO 170
1000 IF CH<>157 AND CH<>158 AND
CH<>159 THEN GOTO 3000
1005 IF D$<>"A" THEN GOTO 3100
1010 LET D$=CHR$ CH
1020 LET TS=CODE D$-156
1030 RETURN
1500 LET SC=SC+TS*10+20
1510 LET D$="A"
1515 LET TEST=0
1520 IF SC=120 THEN GOTO 3000
```

```
1530 PRINT AT OY,OX;" ";AT 2,15;
1540 LET OX=INT (RND*32)
1550 LET OY=INT (RND*10+5)
1560 RETURN
3000 CLS
3005 IF CH>128 THEN PRINT "SIE H
ABEN MIT DER TAUCHERGLOCKE DAS S
CHIFF GERAMMT." AND CH<>999;"NOC
H EINE TRUHE." AND CH=999
3010 IF SC=120 THEN PRINT "ERFOL
G...     ERFOLG..."
3015 IF INT OX=X AND INT OY=Y TH
EN PRINT "GEFANGEN...";
3020 IF CH=CODE "-" THEN PRINT "
SIE SIND AUFGETAUCHT - WARTEN
SIE, BIS SIE GERETTET WERDEN."
3030 PRINT "SIE HABEN ";SC*LEOD+
(100-COUNT);" PUNKTE."
3040 PRINT ,,"UND ";COUNT;" EINH
AEITEN SAUER-";TAB 0;"STOFF VERB
RAUCHT...."
3050 PRINT ,,"NOCH EIN SPIEL? (J
/N) "
3060 LET K$=INKEY$
3070 IF K$="" THEN GOTO 3060
3080 IF K$="J" THEN CLS
3090 IF K$="J" THEN RUN
3095 STOP
3100 LET CH=999
3110 GOTO 3000
9990 STOP
9998 SAVE "KRAKE"
9999 GOTO 1
```

Sternenhändler

Sie sind ein Händler der Sternenwelt und rasen auf Hyperwegen durch die Galaxis, um auf den galaktischen Märkten Umatz zu machen. Beim Aufbruch zur Reise sind Sie mit einem Sortiment von Waren und einer Summe intergalaktischen Geldes ausgestattet. Sie werden versuchen, mit einer größeren Warenmenge zu Ihrem Heimatstern zurückzukehren. Bei Ihrer Rückkehr wird es Ihnen gelingen, diese Waren zum doppelten Handelswert abzusetzen, und als Lohn der Mühe winkt Ihnen dann 10 Prozent vom Reingewinn. Draußen im Weltraum werden Sie andere Händler treffen, die Ihnen An- und Verkaufsgeschäfte zu günstigen Preisen anbieten. Wenn Sie mit bestimmten Partnern keine An- und Verkaufsgeschäfte machen wollen, brauchen Sie nur ENTER zu drücken.
Das Ziel des Spiels ist es, eine möglichst große Menge hochwertiger Waren einzukaufen... Ihr galaktischer Überkaufmann ist Tim Rogers.

```
  10 LET A$="EX-DIKTATOREN,GUMMI
STIEFEL,PAPIERTUETEN,BAUKLOETZE,
SAMENKOERNER,RAKETEN,FUSSBALLSTI
EFEL,BONBONS,SCHMIERSEIFENEIMER,
ORANGENSCHALEN,KILO EIS,WAESCHEK
LAMMERN,REIFEN,ZINK-RADKAPPEN,LI
TER SYNTHETISCHE TINTE,SAECKE KO
HLENSTAUB,METALLBARREN,REAGENZGL
AESER,TUERKNAEUFE,FISCHE,FERNSEH
ER,ABZEICHEN,C-60 KASSETTEN,PANT
OFFEL,BLECHNAEGEL,ZIEGELSTEINE,G
UMMIBAEUME,GLUEHLAMPEN,HANDSCHUH
E,KUGELSCHREIBER"
```

```
  15 FAST
  20 DIM B$(30,25)
  25 LET F=100
  30 DIM L(30)
  40 LET A=1
  45 GOSUB 125
  50 FOR B=1 TO LEN A$
  70 IF A$(B)="," THEN GOSUB F
  80 LET C$=C$+A$(B)
  90 NEXT B
  95 GOTO F
 100 LET B$(A)=C$
 110 LET L(A)=LEN C$
 115 LET A=A+1
 120 LET B=B+1
 125 LET C$=""
 130 IF A<=30 THEN RETURN
 135 IF F=180 THEN GOTO 230
 140 LET A$="20,20,.1,.5,50,1000
,20,.2,.5,1,1,5,10,45,4,5,600,2,
5,5,6,100,1,4,9,3,70,5,6,15,.4"
 150 DIM B(30)
 160 LET F=180
 170 GOTO 40
 180 LET B(A)=VAL C$
 190 GOTO 115
 200 POKE 16389,76
 202 IF M$<>STR$ PP THEN LET M$=
STR$ M
 205 IF LEN M$<2 THEN GOTO 220
 210 IF M$(LEN M$-1)="." THEN LE
T M$=M$+"0"
 220 RETURN
 230 DIM A(30)
 240 LET IA=INT (RND*10000)+1
 250 LET M=IA
 260 FOR B=1 TO 30
 270 IF RND>.85 THEN LET A(B)=IN
T (RND*10000)+1
 280 IF A(B)>0 THEN LET IA=IA+B(
B)*2
 290 NEXT B
```

```
 300 LET Y=2100+INT (RND*3000)
 305 LET PP=0
 310 SLOW
 315 LET M$=STR$ M
 320 GOSUB 200
 330 FOR Y=Y TO Y+20
 340 CLS
 350 PRINT TAB 12;"JAHR ";Y
 355 LET P=3
 360 PRINT "********************
************"
 370 PRINT "GELD $";
 380 FOR B=1 TO LEN M$
 390 PRINT CHR$ (CODE M$(B)+128)
;
 400 NEXT B
 405 PRINT
 410 FOR B=1 TO 30
 420 IF A(B)=0 THEN GOTO 440
 425 PRINT A(B);TAB 6;B$(B)
 430 LET P=P+1
 435 IF P>=18 THEN SCROLL
 440 NEXT B
 450 LET S$="VERKAUFEN"
 460 GOSUB 1000
 470 FOR P=1 TO NT
 475 IF RND<.47 THEN GOTO 900
 480 PRINT AT (P-1)*5,0;"HAENDLE
R";P;
 490 PRINT " MOECHTE ";S$;
 491 IF S$="KAUFE" THEN PRINT "N
"
 500 LET IT=INT (RND*30)+1
 510 LET PP=INT (RND*100)+1
 520 LET NA=INT (RND*1000)+1
 530 LET XA=INT (RND*10000)+NA
 540 LET PP=PP/100*B(IT)
 550 LET PP=INT (PP*100)/100
 560 LET T$=M$
 570 LET M$=STR$ PP
 580 GOSUB 200
 590 LET P$=M$
```

```
 600  LET M$=T$
 670  IF S$="KAUFE" THEN GOSUB 11
10
 680  PRINT XA;" ";B$(IT, TO L(IT
));" ZU JE ","$";P$
 690  PRINT AT 19,0;"WIE VIELE ";
B$(IT, TO L(IT));" ";
 695  IF L(IT)>14 THEN PRINT
 700  IF S$="KAUFE" THEN GOSUB 11
30
 710  PRINT "KAUFEN? (SIE HABEN "
;M$;"$)"
 720  INPUT Z$
 725  IF Z$="" THEN GOTO 830
 730  IF CODE Z$<28 OR CODE Z$>37
 THEN GOTO 720
 740  LET Z=VAL Z$
 745  LET Z=INT (Z+.5)
 750  IF Z<0 OR Z>XA THEN GOTO 72
0
 760  LET CS=Z*PP
 765  IF S$="KAUFE" THEN RETURN
 770  IF M-CS<0 THEN GOTO 720
 780  LET M=M-CS
 790  GOSUB 200
 800  LET A(IT)=A(IT)+Z
 810  CLS
 820  NEXT P
 825  GOTO 860
 830  IF C$="VERKAUFEN" THEN GOTO
 810
 850  GOTO 1185
 860  IF S$="KAUFE" THEN GOTO 120
0
 865  LET S$="KAUFE"
 870  GOSUB 1000
 880  GOTO 470
 900  PRINT AT (P-1)*5,0;"WAS WOL
LEN SIE ";
 910  IF S$="KAUFE" THEN PRINT "V
ERKAUFEN AN "
 920  IF S$="VERKAUFEN" THEN PRIN
```

```
T "KAUFEN VON "
 930 PRINT "HAENDLER ";P;"?"
 940 INPUT Z$
 945 IF Z$="" THEN GOTO 830
 950 LET LZ=LEN Z$
 955 IF LZ<2 THEN GOTO 940
 960 FOR L=1 TO 30
 970 IF B$(L, TO LZ)=Z$ THEN GOT
O 1500
 980 PRINT "MM..";
 990 NEXT L
 995 GOTO 1510
1000 LET NT=INT (RND*5)
1010 PRINT AT 19,0;NT;" HAENDLER
 WOLLEN "
1020 PRINT S$;
1030 IF S$="KAUFE" THEN PRINT "N
 BEI IHNEN"
1040 IF S$="VERKAUFEN" THEN PRIN
T " AN SIE"
1060 PRINT AT 21,0;"BITTE BELIEB
IGE TASTE DRUECKEN"
1070 PAUSE 4E4
1080 POKE 16437,255
1090 CLS
1100 RETURN
1110 PRINT NA;" - ";
1120 RETURN
1130 PRINT "VERKAUFEN? (SIE HABE
N"
1131 PRINT A(IT);" ";B$(IT, TO L
(IT));")"
1140 GOSUB 720
1150 IF Z<NA OR A(IT)<Z THEN GOS
UB 720
1160 LET A(IT)=A(IT)-Z
1170 LET M=M+CS
1180 GOSUB 200
1185 CLS
1190 NEXT P
1200 NEXT Y
1210 LET FA=M
```

```
1220 FOR B=1 TO 30
1230 PRINT AT 11,10;"SIE SIND ZU
HAUSE"
1240 IF A(B)>0 THEN LET FA=FA+B(
B)*2
1250 PRINT AT 11,10;"SIE SIND ZU
HAUSE"
1260 NEXT B
1270 CLS
1280 PRINT "SIE BEGANNEN MIT ";I
A;"$"
1290 PRINT "AN GELD UND LADUNG."
1300 PRINT ,,"JETZT BESITZEN SIE
 ";FA;"$"
1310 PRINT "GELD UND LADUNG."
1320 LET PR=FA-IA
1330 IF SGN PR=-1 THEN LET P$="V
ERLUST"
1340 IF SGN PR>-1 THEN LET P$="G
EWINN"
1350 PRINT ,,"SIE MACHTEN ";ABS
PR;"$";P$
1360 IF P$="GEWINN" THEN PRINT ,
,,,"IHR VERDIENST IST ";INT (PR*
10)/100;"$"
1370 IF P$="VERLUST" THEN PRINT
,,,,"SIE KOMMEN INS SCHULDGEFAEN
GNIS UND WERDEN DAS TAGESLICHT N
IE-   MALS WIEDERSEHEN; GESCHWEIG
E     DENN DAS INNERE EINES RAUMS
CHIF-FES."
1372 PAUSE 200
1375 POKE 16437,255
1380 IF P$="VERLUST" THEN NEW
1390 PRINT ,,,,"IHR ARBEITGEBER
BIETET IHNEN EI-NEN NEUEN VERTRA
G AN.              NEHMEN SIE IHN A
N? (J/N)"
1400 IF INKEY$<>"J" AND INKEY$<>
"N" THEN GOTO 1400
1410 IF INKEY$="N" THEN STOP
1420 FAST
```

```
1430 GOTO 230
1500 CLS
1505 IF RND>.4 THEN GOTO 1600
1510 CLS
1520 IF L<31 THEN LET Z$=B$(L, T
O L(L))
1530 PRINT AT (P-1)*5,0;"HAENDLE
R ";P;
1535 LET R=RND
1540 IF S$="VERKAUFEN" AND R<.5
THEN PRINT " HAT KEINE"
1550 IF S$="VERKAUFEN" AND R>=.5
 THEN PRINT " VERKAUFT KEINE "
1560 IF S$="KAUFE" THEN PRINT "
WILL KEINE"
1570 PRINT Z$
1580 PRINT AT 21,0;"BITTE BELIEB
IGE TASTE DRUECKEN"
1585 PAUSE 4E4
1590 POKE 16437,255
1595 GOTO 830
1600 PRINT AT (P-1)*5,0;"HAENDLE
R ";P;" MOECHTE ";S$;
1602 IF S$="KAUFE" THEN PRINT "N
";
1604 PRINT
1610 LET IT=L
1620 GOTO 510
2010 SAVE "SKLAVEN II"
2020 POKE 16389,128
2030 FAST
2040 GOTO 230
```

Schatzkistenjagd

Jahrelang haben Sie im Bermuda-Dreieck nach verschollenen Schatzkisten gesucht. Endlich stoßen Sie hinter einem gefährlichen Korallenriff auf die ersehnte Beute. Es sind gleich eine ganze Reihe von Schatzkisten, aber sie wissen, daß alle Kisten leer sind – bis auf die letzte, die mit so viel Juwelen und Gold gefüllt ist, daß Sie für immer ausgesorgt haben. Die leeren Kisten müssen Sie erst beiseite schaffen, um an die wahre Schatzkiste zu gelangen. Dummerweise war Ihr erbarmungsloser Konkurrent ZX 81 Ihnen die ganze Zeit auf den Fersen und taucht nun auf, um die Beute an sich zu reißen.
Abwechselnd nehmen Sie und der Computer 1, 2 oder 3 Kisten weg. Sie bestimmen das mit den Tasten "1" bis "3". Der Stand des Spiels wird sowohl in Ziffern wie auch graphisch angezeigt. Gewonnen hat der Spieler, der im entscheidenden Augenblick, wenn alle anderen Kisten weg sind und nur noch die wahre Schatzkiste zu holen ist, zum Zug kommt. Machen Sie sich keine Illusionen! Meistens hat der ZX 81 die besseren Taucher und wird gewinnen. Nur bei äußerster Konzentration können Sie ihn überlisten. Ausgetüftelt wurde diese teuflische Schatzkistenjagd von Dilwyn Jones.

```
  10 RAND
  15 LET P=INT (RND*15)+10
  20 PRINT AT 0,0;"2████████████
███████████"( TO P);AT 5,5;P;" K
ISTEN"
  30 IF RND<.5 THEN GOTO 120
  40 PRINT AT 8,0;"
  "
```

```
  50 PRINT AT 12,0;"WIEVIELE KIS
TEN WOLLEN SIE      NEHMEN?"
  60 LET R=CODE INKEY$-28
  70 IF R<1 OR R>3 THEN GOTO 60
  80 PRINT AT 12,0;"
                             ";AT
9,0;"                    "
  90 LET P=P-R
 100 PRINT AT 5,5;P;" KISTE";"N"
 AND P<>1;" BLEIBEN  ";AT 8,0;"S
IE NAHMEN ";R;
 105 GOSUB 240
 110 IF P<1 THEN GOTO 220
 120 LET N=INT (P/4)*4
 130 IF P=N THEN LET R=INT (RND*
3)+1
 140 IF P<>N THEN LET R=P-N
 150 LET P=P-R
 160 PRINT AT 9,0;"ICH NEHME ";R
;
 165 GOSUB 240
 170 PRINT AT 5,5;P;" KISTE";"N"
 AND P<>1;" BLEIBEN "
 180 IF P<1 THEN GOTO 200
 190 GOTO 40
 200 PRINT AT 0,0;"£££ ICH GEWIN
NE £££"
 210 STOP
 220 PRINT AT 0,0;"£££ SIE GEWIN
NEN £££"
 230 STOP
 240 FOR A=1 TO 10
 250 PRINT AT 0,P;"███"( TO R);A
T 0,P;"             "( TO R)
 260 NEXT A
 270 RETURN
9998 SAVE "SCHATZ"
9999 GOTO 1
```

Bomben Nim

In dieser Nim-Variation, ebenfalls von Dilwyn Jones erdacht, fliegen Sie in die Luft, wenn Sie sich in die Lage manövrieren lassen, das letzte Exemplar in einer ganzen Reihe von Objekten nehmen zu müssen. Bei diesen Objekten handelt es sich nämlich um Bomben, die entschärft werden müssen. Bei allen Bomben gelingt diese Operation, nur bei der letzten können Sie so sensibel vorgehen, wie sie wollen: sie wird auf jeden Fall explodieren. Berufsrisiko!

```
   10 RAND
   15 LET P=INT (RND*15)+10
   20 PRINT AT 0,0;"⊠████████████
███████████"( TO P);AT 5,5;P;" B
OMBEN "
   30 IF RND<.5 THEN GOTO 120
   40 PRINT AT 8,0;"
   "
   50 PRINT AT 12,0;"WIEVIELE BOM
BEN ENTSCHAERFEN SIE"
   60 LET R=CODE INKEY$-28
   70 IF R<1 OR R>3 OR R>P THEN G
OTO 60
   80 PRINT AT 12,0;"
                     ";AT 9,0;"
                "
   90 LET P=P-R
  100 PRINT AT 5,5;P;" BOMBE";"N"
 AND P<>1;" BLEIB";"EN  " AND P<
>1;"T   " AND P=1;AT 8,0;"SIE NA
HMEN ";R;
  105 GOSUB 240
  110 IF P<1 THEN GOTO 200
  120 LET N=INT (P/4)*4
  130 IF P=N+1 THEN LET R=INT (RN
```

```
D*3)+1
 140 IF P<>N+1 THEN LET R=P-(INT
 ((P-1)/4)*4)-1
 145 IF R>P THEN LET R=P
 150 LET P=P-R
 160 PRINT AT 9,0;"ICH ENTSCHAER
FE ";R
 165 GOSUB 240
 170 PRINT AT 5,5;P;" BOMBE";"N"
 AND P<>1;" BLEIB";"EN" AND P<>1
;"T  " AND P=1
 180 IF P<1 THEN GOTO 220
 190 GOTO 40
 200 PRINT AT 0,0;" PENG  ICH GE
WINNE  PENG ";
 205 PRINT AT 12,0;" PENG  ICH G
EWINNE  PENG ";
 210 STOP
 220 PRINT AT 0,0;" PENG -SIE GE
WINNEN- PENG"
 225 PRINT AT 12,0;" PENG -SIE G
EWINNEN- PENG"
 230 STOP
 240 FOR A=1 TO 10
 250 PRINT AT 0,P;"███"( TO R);A
T 0,P;"   "( TO R)
 260 NEXT A
 270 RETURN
9998 SAVE "BOMBE"
9999 GOTO 1
```

```
⊠ █ █ █ █ █ █ █ █ █


     10 BOMBEN BLEIBEN


SIE NAHMEN 3
ICH ENTSCHAERFE 1
```

Poker

Dieses Programm spielt nicht nur Poker, sondern druckt auch die Karten aus. Das verlängert das Spiel zwar erheblich, macht es aber auch sehr viel vergnüglicher. Das Display zeigt Ihre Poker-Hand und die des Computers, als zwei Sets von je fünf Karten. Die Karten des Computers sehen Sie natürlich nur von hinten!
Zunächst werden Sie gefragt, wieviel Sie setzen wollen.

Die Wertigkeit der Karten von der niedrigsten zur höchsten Hand:
Nichts – verliert gegen alles folgende
1 Paar – Chancen 1:2
2 Paare – Chancen 1:21
3 Gleiche – Chancen 1:47
Straight – Chancen 1:255
Flush – Chancen 1:509
Full House – Chancen 1:694
4 Gleiche – Chancen 1:4.165
Straight Flush – Chancen 1:72.193
Royal Flush – Chancen 1:169.740

Nach Ihrem ersten Einsatz können Sie bis zu drei neue Karten nehmen. Dasselbe darf der Computer. Der Computer deckt nun seine Karten auf und vergleicht die beiden Hände. Wenn Sie Ihr ganzes Geld verlieren, können Sie sich etwas vom Computer leihen und bei einer neuen Runde versuchen, es zurückzugewinnen. Sie können das Programm speichern, indem Sie den Recorder starten und einfach auf RUN und ENTER drücken; das Programm wird dann automatisch beim Loading gestar-

tet. Löschen Sie Zeile 1, wenn Sie von dieser Möglichkeit keinen Gebrauch machen wollen.
Die Zeilen 1 bis 1Ø5 stellen die Variablen auf. Die Zeilen 11Ø bis 138 stellen das Spiel vor. Die Zeilen 14Ø bis 592 erklären die Spielregeln.

```
   1 SAVE "POKER"
   2 LET S$="
                "
   5 LET BORROW=0
  10 RAND
  12 DIM A$(9,5)
  14 DIM T$(11,5)
  16 LET A$(1)="█▀▀▀█ "
  18 LET A$(2)="█▄▄▄█ "
  20 LET A$(3)="█▪▪▪█ "
  22 LET A$(4)="█▪ ▪█ "
  24 LET A$(5)="█▪ ▪█ "
  26 LET A$(6)="█ ▪ █ "
  28 LET A$(7)="█ ▪ █ "
  30 LET A$(8)="█   █ "
  32 LET A$(9)="█ ██ █ "
  34 LET T$(1)="99992"
  36 LET T$(2)="68682"
  38 LET T$(3)="76872"
  40 LET T$(4)="48842"
  42 LET T$(5)="46842"
  44 LET T$(6)="45842"
  46 LET T$(7)="35842"
  48 LET T$(8)="44442"
  50 LET T$(9)="43442"
  52 LET T$(10)="34342"
  54 LET T$(11)="86882"
 100 DIM T(13)
 101 DIM K(4)
 102 DIM M(5)
 103 DIM N(5)
 104 DIM C(5)
 105 DIM D(5)
 110 PRINT TAB 8;"K< POKER SPIEL
```

```
  > >"
 111 LET P9=100
 112 PRINT
 114 PRINT "SIE SPIELEN EINE PAR
TIE POKER    GEGEN MICH."
 116 PRINT "DER MINDESTEINSATZ I
N JEDER      RUNDE BETRAEGT $5.00
 UND SIE     BEGINNEN MIT $100. I
N JEDER      RUNDE KOENNEN SIE: -
"
 118 PRINT ,,"ERHOEHEN - UM IHRE
N EINSATZ ZU             ERHOEHE
N, JEDOCH                NICH UM
 MEHR ALS DEN            ANFANGS
EINSATZ."
 122 PRINT ,,"SEHEN    - UM DIE
KARTEN DES               COMPUTE
RS ZU SEHEN."
 128 PRINT ,,"AUFLEGEN - UM IHRE
 KARTEN                  AUFZUDE
CKEN."
 129 POKE 16418,0
 130 PRINT ,,"SPIELBEGINN: BITTE
 EINE                       BELIE
BIGE TASTE                  DRUEC
KEN."
 132 PAUSE 4E4
 133 CLS
 134 PRINT AT 10,8;"VIEL GLUECK.
"
 135 PAUSE 100
 138 CLS
 139 POKE 16418,2
 140 DIM S(4,13)
 151 PRINT AT 9,0;"DIE KARTEN SI
ND GEMISCHT."
 160 LET A9=5
 163 PRINT AT 11,9;"EINSATZ $5.0
0"
 165 PRINT AT 13,9;"SIE HABEN $"
;P9
 166 IF P9<=0 THEN GOTO 7000
```

```
 170  FOR X=1 TO 5
 171  GOSUB 920
 172  LET M(X)=C
 173  LET N(X)=S
 174  NEXT X
 175  GOSUB 1040
 181  PRINT ,,"ICH NEHME MEINE KA
RTEN..."
 185  FOR X=1 TO 5
 190  PRINT "▒▒▒▒▒ ▒▒▒▒▒ ▒▒▒▒▒ ▒▒▒
▒▒▒ ▒▒▒▒▒ "
 191  GOSUB 920
 192  LET C(X)=C
 193  LET D(X)=S
 194  NEXT X
 195  PRINT "▒▒▒▒▒ ▒▒▒▒▒ ▒▒▒▒▒ ▒▒▒
▒▒▒ ▒▒▒▒▒  ▒▒▒▒▒ ▒▒▒▒▒ ▒▒▒▒▒ ▒▒▒
▒▒▒ ▒▒▒▒▒ "
 196  PRINT AT 19,0;"SIE HABEN  "
;P9-5;" £"
 200  GOSUB 1080
 201  LET C1=T
 202  LET C2=H9
 203  LET P1=-5
 211  PRINT AT 21,0;"EROEFFNEN SI
E MIT £ 0.00-";P9-5
 212  INPUT BET
 213  IF BET<0 OR BET>P9-5 THEN G
OTO 211
 214  LET B=BET
 216  IF RND<.8 THEN GOTO 230
 220  IF INT (((T*RND)+1)+((H9*RN
D)+1)+(T*10))<BET-(BET/10) THEN
GOTO 440
 230  LET A9=A9+B
 231  GOSUB 5025
 232  PRINT "ICH NEHME AN."
 233  PRINT AT 19,10;P9-5-BET;"
"
 234  LET STAKE=BET
 240  GOSUB 5000
 241  PRINT "WIEVIELE NEUE KARTEN
```

```
?"
 242 IF INKEY$<>"" THEN GOTO 242
 243 IF INKEY$="" THEN GOTO 243
 244 LET E$=INKEY$
 245 IF E$<"0" OR E$>"3" THEN GO
TO 242
 246 LET K9=VAL E$
 250 FOR X=1 TO K9
 260 GOSUB 5025
 261 PRINT "KARTENZAHL.?" AND (X
/2=INT (X/2));"KARTENZAHL?" AND
(X/2<>INT (X/2));"  ";
 262 IF INKEY$<>"" THEN GOTO 262
 263 IF INKEY$="" THEN GOTO 263
 264 LET E$=INKEY$
 265 IF E$<"0" OR E$>"9" THEN GO
TO 262
 266 LET T9=VAL E$
 267 PRINT T9
 268 GOSUB 920
 270 IF T9<6 THEN GOTO 280
 271 GOSUB 5000
 272 PRINT "KARTE NR. (NUR 1-5).
"
 273 GOTO 260
 280 LET M(T9)=C
 281 LET N(T9)=S
 282 NEXT X
 283 GOSUB 1044
 290 GOSUB 1070
 291 FOR X=1 TO 5
 292 LET T(M(X))=T(M(X))+1
 293 LET K(N(X))=K(N(X))+1
 294 NEXT X
 295 GOSUB 5025
 296 PRINT "ICH DENKE.........."
 301 GOSUB 640
 302 LET P1=T
 303 GOSUB 800
 304 LET P2=H9
 311 GOSUB 1080
 312 LET H9=0
```

```
 313 IF T>3 THEN GOTO 350
 320 FOR Z=1 TO 5
 321 IF H9=3 THEN GOTO 340
 322 IF T(C(Z))<>1 THEN GOTO 340
 323 LET H9=H9+1
 330 GOSUB 920
 331 LET C(Z)=C
 332 LET D(Z)=S
 340 NEXT Z
 350 GOSUB 5025
 351 PRINT "ICH NEHME ";H9;" KAR
TE";"N" AND H9>1;"."
 352 LET H8=H9
 360 GOSUB 1080
 361 LET C1=T
 362 GOSUB 800
 363 LET C2=H9
 370 LET B9=INT ((C1*RND)+(C2*RN
D)+(C1*10))+INT (A9/3)+((K9-H8)*
2)+7
 380 LET B1=0
 382 IF RND<C1*.5 THEN GOTO 390
 383 LET B9=99.99
 390 GOSUB 5000
 391 GOSUB 6000
 401 IF B<0 THEN GOTO 570
 402 LET A9=A9+B1
 410 IF B<>0 THEN GOTO 420
 411 GOSUB 480
 412 GOTO 540
 420 IF A9+B<B9 THEN GOTO 460
 421 IF A9+B<B9+(B9/2) THEN GOTO
 450
 430 IF B9=99.99 THEN GOTO 450
 440 GOSUB 5000
 441 PRINT ".........ICH BIN DRA
USSEN......"
 442 GOSUB 480
 443 GOTO 580
 450 GOSUB 5000
 451 PRINT "IHRE KARTEN."
 452 LET A9=A9+B
```

```
453 GOSUB 480
454 GOTO 540
460 LET B1=INT (((B9-A9)/3)*RND
)+2
461 LET A9=A9+B
470 GOSUB 5000
471 PRINT "ICH SETZE $ ";B1
472 GOTO 390
480 GOSUB 4000
481 GOSUB 1044
482 GOSUB 5000
483 PRINT "ICH HATTE ";
484 LET T=C1
485 GOSUB 820
490 GOSUB 5000
491 PRINT "MIT EINER HOHEN HAND
VON ";
492 LET C=C2
493 GOSUB 940
494 PRINT "."
500 IF P1=-5 THEN GOTO 530
510 GOSUB 5000
511 PRINT "SIE HATTEN ";
512 LET T=P1
513 GOSUB 820
520 GOSUB 5000
521 PRINT "MIT EINER HOHEN HAND
VON ";
522 LET C=P2
523 GOSUB 940
524 PRINT "."
531 RETURN
540 IF C1=P1 THEN GOTO 550
541 IF C1<P1 THEN GOTO 580
542 GOTO 570
550 IF C2=P2 THEN GOTO 560
551 IF C2<P2 THEN GOTO 580
552 GOTO 570
560 GOSUB 5000
561 PRINT "UNENTSCHIEDEN ....."
562 GOTO 590
570 GOSUB 5000
```

```
 571  PRINT "ICH HABE GEWONNEN $
";A9
 572  LET P9=P9-A9
 573  GOTO 590
 580  GOSUB 5000
 581  PRINT "SIE HABEN GEWONNEN $
";A9
 582  LET P9=P9+A9
 590  GOSUB 5000
 592  PRINT "NOCH EINMAL (J/N) ?"
 593  IF INKEY$<>"" THEN GOTO 593
 594  IF INKEY$="" THEN GOTO 594
 595  IF INKEY$="J" THEN GOTO 138
 596  CLS
 597  LET P9=P9-(BORROW*100)
 598  SCROLL
 599  PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒"
 601  IF P9<0 THEN GOTO 630
 610  IF P9>=100 THEN GOTO 620
 611  SCROLL
 612  PRINT "SIE VERLIESSEN DEN T
ISCH MIT $ "
 613  SCROLL
 614  PRINT "            ";100-P9
 615  SCROLL
 616  SCROLL
 617  PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒"
 618  GOTO 635
 619  SCROLL
 621  PRINT "SIE HABEN DEN TISCH
MIT EINEM    GEWINN VON $ ";P9;"
VERLASSEN"
 622  SCROLL
 623  SCROLL
 624  PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒"
 625  GOTO 635
 630  SCROLL
 631  PRINT "SIE SCHULDEN MIR $ "
;-P9
```

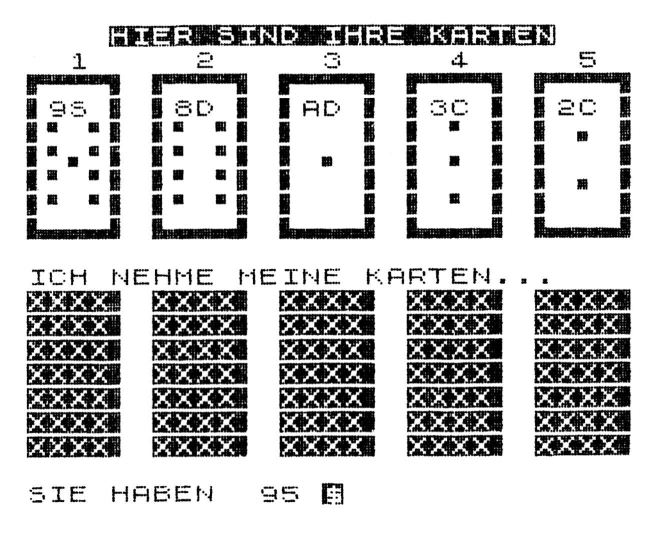
```
632 SCROLL
633 SCROLL
634 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
635 SCROLL
636 SCROLL
637 SCROLL
638 STOP
640 LET T=0
641 FOR I=1 TO 4
642 IF K(I)<>5 THEN GOTO 650
643 LET T=5
650 NEXT I
651 LET I=2
652 LET H9=0
660 LET I=I-1
661 IF I<>0 THEN GOTO 670
662 LET I=13
670 IF T(I)<1 THEN GOTO 660
671 LET H9=I
672 IF I<>1 THEN GOTO 680
673 LET I=14
680 LET Z=I-4
690 LET I=I-1
691 IF T(I)<>1 THEN GOTO 720
700 IF Z<>1 THEN GOTO 690
710 LET T=T+4
711 IF Z<>10 THEN GOTO 720
712 IF T=4 THEN GOTO 170
713 LET T=T+1
720 IF I=13 THEN GOTO 660
721 IF T<>5 THEN GOTO 730
722 IF T(1)<>1 THEN GOTO 730
723 LET H9=1
730 IF T=0 THEN GOTO 740
731 RETURN
740 FOR I=1 TO 13
741 IF T(I)<>4 THEN GOTO 750
742 LET T=7
750 IF T(I)<>3 THEN GOTO 760
751 LET T=T+5
752 LET H9=I
```

```
760 IF T(I)<>2 THEN GOTO 790
761 IF T=5 THEN GOTO 780
762 IF T<>0 THEN GOTO 770
763 LET H9=0
770 IF H9=1 THEN GOTO 780
771 IF H9>I THEN GOTO 780
772 LET H9=I
780 LET T=T+1
790 NEXT I
791 IF T<>5 THEN GOTO 810
792 LET T=3
793 RETURN
800 IF H9<>1 THEN GOTO 810
801 LET H9=14
810 RETURN
820 IF T<>0 THEN GOTO 830
821 PRINT "NICHTS"
822 RETURN
830 IF T<>1 THEN GOTO 840
831 PRINT "EIN PAAR"
832 RETURN
840 IF T<>2 THEN GOTO 850
841 PRINT "ZWEI PAARE"
842 RETURN
850 IF T<>3 THEN GOTO 860
851 PRINT "DREI GLEICHE"
852 RETURN
860 IF T<>4 THEN GOTO 870
861 PRINT "EINEN STRAIGHT"
862 RETURN
870 IF T<>5 THEN GOTO 880
871 PRINT "EINEN FLUSH"
872 RETURN
880 IF T<>6 THEN GOTO 890
881 PRINT "EIN FULL HOUSE"
882 RETURN
890 IF T<>7 THEN GOTO 900
891 PRINT "VIER GLEICHE"
892 RETURN
900 IF T<>9 THEN GOTO 910
901 PRINT "EINEN STRAIGHT FLUSH
"
```

```
 902 RETURN
 910 PRINT "EINEN ROYAL FLUSH...
"
 911 RETURN
 920 LET S=INT (RND*4+1)
 921 LET C=INT (RND*13+1)
 930 IF S(S,C)=1 THEN GOTO 920
 931 LET S(S,C)=1
 932 RETURN
 940 PRINT "A" AND (C=1 OR C=14)
;"T" AND C=10;"J" AND C=11;"Q" A
ND C=12;"K" AND C=13;
 941 IF C>1 AND C<10 THEN PRINT
C;
 942 IF C=1 THEN LET C=14
 943 IF C>10 AND C<14 THEN LET C
=1
 944 IF C=14 THEN LET C=11
 952 RETURN
1000 PRINT "H" AND S=1;"S" AND S
=2;"D" AND S=3;"C" AND S=4;
1001 PRINT " █ ";
1002 RETURN
1040 CLS
1041 LET LINE=0
1042 PRINT TAB 4;"HIER SIND IHRE
 KARTEN"
1043 PRINT "   1      2      3
4     5"
1044 FOR X=1 TO 5
1045 PRINT AT 2+LINE,0;
1046 PRINT TAB (X-1)*6;"▛▀▀▀▀▜ ";T
AB (X-1)*6;"▌ ";
1050 LET C=M(X)
1051 LET S=N(X)
1054 GOSUB 940
1055 GOSUB 1000
1056 FOR Y=1 TO 5
1057 PRINT TAB (X-1)*6;A$(VAL T$
(C,Y))
1058 NEXT Y
1060 NEXT X
```

```
1061 RETURN
1070 DIM K(4)
1071 DIM T(13)
1072 RETURN
1080 GOSUB 1070
1081 FOR X=1 TO 5
1082 LET T(C(X))=T(C(X))+1
1090 LET K(D(X))=K(D(X))+1
1091 NEXT X
1092 GOSUB 640
1093 RETURN
1100 STOP
4000 FOR Q=1 TO 5
4010 LET M(Q)=C(Q)
4020 LET N(Q)=D(Q)
4030 NEXT Q
4040 LET LINE=9
4050 RETURN
5010 FOR Q=1 TO 25
5020 NEXT Q
5025 PRINT AT 21,0;S$;AT 21,0;
5030 RETURN
6000 GOSUB 5000
6005 PRINT "  ERHOEHEN SEHEN AUS
STEIGEN ?"
6010 IF INKEY$<>"" THEN GOTO 601
0
6020 IF INKEY$="" THEN GOTO 6020
6030 LET E$=INKEY$
6040 IF E$<>"E" AND E$<>"S" AND
E$<>"A" THEN GOTO 6010
6050 IF E$="S" THEN LET B=0
6060 IF E$="A" THEN LET B=-1
6065 IF E$="S" OR E$="A" THEN GO
TO 6120
6070 GOSUB 5025
6080 PRINT "    UM WIEVIEL ERHOEH
EN SIE ?"
6090 INPUT B
6100 IF B>BET OR B+STAKE>P9 THEN
 GOTO 6090
6110 IF B<=0 THEN GOTO 6000
```

```
6114 LET STAKE=STAKE+B
6115 PRINT AT 19,10;P9-STAKE-5;"
    "
6120 RETURN
7000 PRINT
7010 PRINT "        SIE HABEN KEIN
 GELD"
7011 PRINT " WOLLEN SIE 100$ LEI
HEN (J/N) ?"
7020 IF INKEY$<>"" THEN GOTO 702
0
7030 IF INKEY$="" THEN GOTO 7030
7040 LET E$=INKEY$
7050 IF E$<>"J" THEN GOTO 596
7060 LET P9=100
7065 LET BORROW=BORROW+1
7070 GOTO 138
```



Nessie

In diesem Spiel, das von Dilwyn Jones erdacht wurde und nicht mehr als 1K braucht, versteckt sich das Monster von Loch Ness hinter einem Felsen auf dem schaurigen Grunde des Sees, wo es, damit die Aufgabe auch richtig schwierig wird, viele solcher Felsen gibt. Sie müssen also erst einmal herausfinden, hinter welchem Felsen sich die schreckliche Schlange verbirgt. Zugunsten der Suchaktion sind die Felsen ordentlich numeriert. Drückt man die Taste mit der Zahl, die der Zahl auf einem Felsen entspricht, so zerbirst dieser Fels. War es nicht der Fels, hinter dem das Monster sich versteckt, so passiert gar nichts, außer daß das Spiel weitergeht. Schließlich gelangt man an den Felsen mit dem Monster: der Fels zerbirst, der Schlamm wird aufgewühlt, das Monster erscheint und schickt sich sogleich an, sich auf Sie zu stürzen und Sie aufzufressen. Um das zu verhindern, müssen Sie eine Buchstaben-Taste drücken und so Nessie den Garaus machen.
Die Zeile 10 erzeugt eine Zahl von Eins bis Neun und verwandelt sie in ein String-Gebilde mit der STR$-Funktion, zum Zweck des späteren Kontroll-Vergleichs mit der auf dem Keyboard gedrückten Taste. Die invers gedruckten Zahlen sind die Felsen (graphisch eindrucksvollere Felsen konnten wir leider nicht zuwege bringen). Die Zeile 30 überwacht das Keyboard. Die Zeile 40 hält das Programm davon ab, weiterzulaufen, bevor eine Zahl von 1 bis 9 gedrückt ist. Die Zeile 50 erzeugt den Raum, der an die Stelle des zerstörten Felsens getreten ist. Die Zeilen 60 bis 80 verursachen eine Verzögerung,

darstellend die Zeit, die es braucht, bis der aufgewühlte Bodenschlamm sich wieder gelegt hat. Während dieser Verzögerung dürfen Sie keine Buchstaben-Taste drükken.

Wenn Sie einen Felsen zerstört haben und Nessie kommt nicht zum Vorschein, verhilft Ihnen die Zeile 90 dazu, es noch einmal zu versuchen. Wenn Sie endlich den richtigen Felsen erwischt haben, setzt Zeile 100 das Programm fort und bringt das Monster ins Spiel. Die Schleifen in den Zeilen 110 bis 130 geben dem Spieler die befristete Chance, eine Buchstaben-Taste zu drücken und so das Monster anzugreifen. Verpaßt der Spieler diese Chance, so schlägt Nessie ihrerseits zu. Wenn Sie noch einmal auf Nessie-Jagd gehen wollen, geben Sie ein "Y" ein.

```
   1 RAND
  10 LET U$=STR$ (INT (RND*9)+1)
  20 PRINT AT 3,3;"1  2  3  4  5
 6  7  8  9";AT 8,14;"   ";AT 9
,14;"   "
  30 LET A$=INKEY$
  40 IF A$<"1" OR A$>"9" THEN GO
TO 30
  50 PRINT AT 3,VAL A$*3;" "
  60 FOR I=1 TO RND*50+10
  70 IF INKEY$>"9" THEN GOTO 210
  80 NEXT I
  90 IF A$<>U$ THEN GOTO 30
 100 PRINT AT 3,VAL U$*3;"*"
 110 FOR I=1 TO 8
 120 IF INKEY$>"9" THEN GOTO 190
 130 NEXT I
 140 PRINT AT 8,14;"YUMYUM";T
AB 14;"   ","OOPS, EIN BISSCHEN
LANGSAM HEUTE"
 145 PRINT AT 12,0;"NOCH EINMAL
```

```
?"
 150 INPUT A$
 160 IF CODE A$<>CODE "J" THEN S
TOP
 170 CLS
 180 RUN
 190 PRINT AT 3,(VAL U$*3-2);"ER
WISCHT"
 200 GOTO 145
 210 PRINT AT 0,7;"ZU FRUEH AM A
BZUG"
 220 GOTO 145
 230 SAVE "NESSI█"
 240 RUN
```

Camel

Sie mußten mitten in der Sahara landen und irren nun durch den Wüstenstaub, nordwärts, südwärts, ostwärts und westwärts, je nachdem, ob Sie die Taste N (North), S (South), E (East) oder W (West) drücken. Auf ihren Irrwegen finden Sie Schätze, aber Sie müssen auch Giftschlangen, Wasserdieben und anderen schrecklichen Gefahren ausweichen. Insgesamt warten 10 Bedrohungen und Schätze auf Sie. Die Spielanleitung ist im Programm enthalten. „Camel" wurde von Chris Callender geschrieben und von Chris Gifford adaptiert.

```
   5 LET T$=""
  10 PRINT TAB 12;"▛▗▝▗▄▝▗▝▗▜"
  15 PRINT TAB 12;"▌CAMEL ▐"
  20 PRINT TAB 12;"▙▘▗▘▀▗▘▗▘▟"
  25 PRINT
  30 PRINT
  35 PRINT "HALLO, WIE HEISSEN S
IE ?"
  40 INPUT N$
  45 PRINT
  50 PRINT "HALLO, ";N$;"."
  55 PRINT
  60 PRINT "KENNEN SIE DIE REGEL
N ? (J/N)"
  65 INPUT R$
  70 IF R$(1)="N" THEN GOSUB 800
0
  75 LET W=200
  80 DIM R(200)
  85 FOR F=1 TO 10
  90 LET R(F)=F*500+500
  95 NEXT F
  96 FOR F=11 TO 200
  97 LET R(F)=R(F-10)
```

```
 100 NEXT F
 105 DIM P$(200,25)
 110 DIM P(200,2)
 115 DIM S$(20,32)
 120 FOR B=1 TO 100
 125 LET P(B,1)=INT (RND*27)+2
 130 LET P(B,2)=INT (RND*17)+2
 135 NEXT B
 140 LET P$(1)="EIN WIRBELNDER S
ANDSTURM"
 145 LET P$(2)="EINE GIFTIGE SCH
LANGE"
 150 LET P$(3)="EINE GROSSE OASE
"
 155 LET P$(4)="DIE VERLORENEN J
UWELEN"
 160 LET P$(5)="EIN KAMEL"
 165 LET P$(6)="EIN ZEITSPRUNG"
 170 LET P$(7)="EIN COMPUTER"
 175 LET P$(8)="EINE KISTE GOLD"
 180 LET P$(9)="EIN SCORPION"
 185 LET P$(10)="WASSERDIEBE"
 190 FOR F=11 TO 200
 195 LET P$(F)=(P$(F-10))
 200 NEXT F
 205 LET X=INT (RND*27)+2
 210 LET Y=INT (RND*18)+2
 215 LET S$(1)="1   5    10   15
    20   25   30 "
 220 LET S$(20)="20

"
 225 FOR B=2 TO 19
 230 LET S$(B)="█
                █"
 232 NEXT B
 234 FOR F=5 TO 19 STEP 5
 236 LET S$(F,1)="5"
 238 NEXT F
 240 LET B$="█"
 245 FOR B=2 TO X-1
 250 LET B$=B$+" "
 255 NEXT B
```

```
 260 LET B$=B$+"*"
 265 FOR B=X+1 TO 31
 270 LET B$=B$+" "
 275 NEXT B
 280 LET B$=B$+"█"
 285 LET S$(Y)=B$
 290 LET S$(18)="█
                A  █"
 295 CLS
 300 FOR B=1 TO 20
 305 PRINT S$(B)
 310 NEXT B
 315 LET N=1
 320 IF P(N,1)=ABS X AND P(N,2)=
ABS Y THEN GOTO 420
 325 LET N=N+1
 330 IF N<>201 THEN GOTO 320
 335 IF ABS Y=18 AND ABS X=29 TH
EN GOTO 7000
 340 PRINT "RICHTUNG ?"
 345 INPUT D$
 350 LET C$=D$(2 TO (LEN D$))
 355 LET C=VAL (C$)
 360 IF C>10 THEN GOTO 345
 365 LET D$=D$(1)
 370 IF D$="N" THEN LET Y=Y-C
 375 IF D$="S" THEN LET Y=Y+C
 380 IF D$="O" THEN LET X=X+C
 385 IF D$="W" THEN LET X=X-C
 390 IF X=30 AND Y=19 THEN GOTO
7000
 395 IF W<10 THEN PRINT "HOLEN S
IE SCHNELL WASSER"
 400 PAUSE 100
 405 LET W=W-1
 410 IF W=0 THEN GOTO 6000
 415 GOTO 215
 420 GOTO R(N)
1000 CLS
1005 PRINT "SIE KOMMEN IN EINEN
SANDSTURM.  SIE SEHEN NICHTS MEH
R UND KOMMENNACH";
```

```
1015 LET X=INT (RND*28)+2
1020 LET Y=INT (RND*19)+2
1025 PRINT "(";X;",";Y;")"
1030 PAUSE 100
1035 GOSUB 9050
1040 GOTO 215
1500 CLS
1505 PRINT "OH VERDAMMT; EINE SC
HLANGE BRRR ICH HASSE SCHLANGEN"
1510 PRINT "ERSCHLAGEN SIE SIE B
ITTE"
1515 LET A=INT (RND*5)
1520 IF A=1 THEN GOTO 1550
1525 PRINT "OH NEIN - SIE HAT SI
E GEBISSEN"
1530 PAUSE 200
1531 CLS
1532 PRINT "NOCH EIN SPIEL (J/N)
 ?"
1535 INPUT G$
1540 IF G$(1)="J" THEN RUN
1545 GOTO 9999
1550 PRINT "GLUECK GEHABT....VOR
SICHTIG SEIN"
1555 GOSUB 9050
1560 PAUSE 250
1565 GOTO 215
2000 PRINT "GROSSARTIG - DA IST
WASSER"
2010 LET A=INT (RND*2)+1
2015 IF A=2 THEN GOTO 2060
2020 PRINT "SIE BEKOMMEN 100 PUN
KTE"
2030 LET W=W+100
2040 GOTO 2065
2050 GOTO 9999
2060 PRINT "NEIN - SIE WURDEN GE
TAEUSCHT"
2065 PAUSE 250
2070 GOSUB 9050
2075 GOTO 215
2500 PRINT "GROSSARTIG HIER SIND
```

```
 DIE            VERSCHOLLENEN JUWELE
N"
2510 LET T$=T$+"DIE VERSCHOLLENE
N JUWELEN"
2520 GOSUB 9000
2525 PAUSE 250
2530 GOTO 215
3000 PRINT "GUT ";N$;" HIER IST
EIN KAMEL"
3004 PAUSE 250
3006 CLS
3010 PRINT "WOHIN SOLL ES SIE TR
AGEN ?"
3012 PRINT "GEBEN SIE X , NEUEIN
E DANN Y EIN"
3020 INPUT X
3025 INPUT Y
3030 GOSUB 9050
3035 GOTO 215
3500 PAUSE 250
3505 CLS
3510 FOR B=1 TO 10
3515 PRINT "   ZEITSPRUNG     ZEI
TSPRUNG      ";
3517 PRINT "   ZEITSPRUNG     ZEI
TSPRUNG"
3520 NEXT B
3525 PAUSE 250
3530 CLS
3535 GOTO 1530
4000 CLS
4003 PRINT "HIER ";N$;" IST EIN
COMPUTER"
4005 PRINT "ER HILFT IHNEN SUCHE
N NACH:"
4006 PAUSE 200
4008 CLS
4010 PRINT "SUCHE NACH .........
............:"
4015 FOR B=1 TO 10
4020 PRINT B;"  ";P$(B);CHR$ 118
4025 NEXT B
```

```
4027 PAUSE 300
4028 CLS
4030 PRINT "WELCHES WOLLEN SIE ?
"
4035 INPUT Z
4040 IF P(Z,1)<>0 AND P(Z,2)<>0
THEN GOTO 4065
4045 PRINT "ES GIBT KEINEN ";P$(
Z)
4050 PAUSE 200
4055 CLS
4060 GOTO 4020
4065 PRINT "VERSUCHE:-(";P(Z,1);
",";P(Z,2);")"
4070 PAUSE 200
4075 GOSUB 9050
4080 GOTO 215
4500 PRINT "HIER ";N$;" IST EINE
 KISTE GOLD"
4510 LET T$=T$+"EINE KISTE GOLD"
4515 GOSUB 9000
4520 PAUSE 200
4525 GOTO 215
5000 PRINT "DA IST EIN SCORPION
";N$
5005 LET A=INT (RND*2+1)
5010 IF A=1 THEN GOTO 1525
5015 PRINT "GUT - ER HAT SIE NIC
HT GEBISSEN"
5020 GOSUB 9050
5025 PAUSE 200
5030 GOTO 215
5500 PRINT "ICH SEHE EINEN WASSE
RDIEB"
5505 LET A=INT (RND*3)
5510 IF A<>1 THEN GOTO 5535
5515 PRINT "SIE HABEN IHN UEBERW
AELTIGT"
5520 PAUSE 200
5525 GOSUB 9050
5530 GOTO 215
5535 LET Q=INT (RND*W)
```

```
5540 PRINT "ER HAT ";Q;" LITER G
ESTOHLEN"
5545 LET W=W-Q
5550 PAUSE 200
5552 GOSUB 9000
5560 GOTO 120
6000 PRINT "SIE SIND VERDURSTET
";N$
6005 PRINT "HABEN SIE ES SCHON G
EMERKT ?"
6010 PAUSE 200
6015 CLS
6020 GOTO 1530
7000 PRINT "SIE SIND AUS MEINER
WUESTE          LEBEND ENTKOMMEN"
7010 IF T$="" THEN GOTO 7040
7020 PRINT "UND KONNTEN MITNEHME
N :-"
7022 PAUSE 200
7025 CLS
7030 PRINT T$
7040 PRINT "BEREITEN SIE SICH AU
F IHREN         FLUG VOR"
7045 GOTO 1530
8000 CLS
8002 PRINT
8003 PRINT
8010 PRINT "WILLKOMMEN ZUM CAMEL
 SPIEL"
8020 PRINT "DAS ZIEL VON CAMEL I
ST; SICH MITDEN TASTE 7 5 8 6 DU
RCH DIE         WUESTE ZU BEWEGEN"
8030 PRINT "GEBEN SIE NACH DER R
ICHTUNG EINEZAHL UNTER 5 EIN, UM
 DIE LAENGE IHRER SCHRITTE FESTZ
ULEGEN."
8040 PRINT "85= SIE GEHEN 5 MEIL
EN OESTLICH"
8050 PRINT "IN DER WUESTE GIBT E
S 10            VERBORGENE SCHAETZE
UND GEFAHREN"
8060 PRINT "SIE SOLLEN SO VIEL S
```

```
CHAETZE WIE MOEGLICH SAMMELN, UN
D ZUM FLUG- HAFEN KOMMEN, DER DA
S A IN DER  RECHTEN ECKE IST"
8070 PRINT "MEIDEN SIE REIHE 18
DORT IST EINSTARKES HITZEFELD, D
AS IHNEN DIESICHT UEBER IHRE LAG
E UNMOEGLICH MACHT."
8075 PRINT
8080 PRINT "*** VIEL SPASS IN DE
R WUESTE ***"
8160 PAUSE 1500
8165 RETURN
9000 IF N<=10 THEN GOTO 9015
9005 LET N=N-10
9010 GOTO 9000
9015 FOR F=N TO 200 STEP 10
9020 LET P(F,1)=0
9025 LET P(F,2)=0
9030 NEXT F
9035 RETURN
9050 LET P(N,1)=0
9055 LET P(N,2)=0
9060 RETURN
9070 SAVE "CAME█"
9080 RUN
9999 STOP
```

Höhlen

Nur wenn Sie sich auf das Äußerste konzentrieren, entgehen Sie bei diesem Programm der Chance, völlig entnervt zu werden. Von Paul Toland erfunden und zu einem herrlichen Graphics-Spiel gestaltet, ist dieses Spiel eine Übung in mentaler Arithmetik, angewandt auf ein spezifisches Problem. Sie müssen den Ausgang eines Höhlen-Labyrinths finden, wobei Sie die Route frei wählen dürfen. Das teuflische Problem liegt darin, daß Sie unterwegs gelegentlich auf zwei mysteriöse Gesamtsummen stoßen, A und B, vermehrt oder vermindert um Zahlen, die im Display angezeigt sind. Sie müssen herausfinden, wie Sie aus den As und Bs wieder Nullen machen, nur dann können Sie Ihren Weg sicher fortsetzen. Noch ein paar zusätzliche Schikanen: wenn Sie gegen eine Wand stoßen oder auf eines der inversen X, müssen Sie zurück an den Start. Am Ende des Spiels bietet man Ihnen eine weitere Runde an. Wenn Sie es noch einmal versuchen wollen, geben Sie ENTER ein – wenn nicht, ein N.

```
   5 CLS
  10 GOSUB 430
  12 PRINT "░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░
░░░░░░░░░░░░░░░░░░░░"
  13 LET A=0
  14 LET B=A
  15 LET T=A
  20 LET X=2
  25 LET N=0
  30 LET Y=1
  40 LET P=PEEK 16396+PEEK 16397
*256+1
  50 PRINT AT Y,X;" "
```

```
  60 LET X1=X+(INKEY$="8")-(INKE
Y$="5")
  70 LET Y1=Y+(INKEY$="6")-(INKE
Y$="7")
  80 PRINT AT Y,X;CHR$ N
  90 LET N=PEEK (P+Y1*33+X1)
 100 PRINT AT Y1,X1;"O"
 110 IF N=0 THEN GOTO 210
 115 IF N<166 OR N>191 THEN GOTO
 20
 120 IF N=189 THEN GOTO 310
 130 IF N=179 THEN GOTO 260
 140 LET A=A+S(N-165,1)
 150 LET B=B+S(N-165,2)
 160 LET D$=STR$ A
 170 PRINT AT 1,28;D$
 180 LET D$=STR$ B
 190 PRINT AT 3,28;D$
 210 LET X=X1
 220 LET Y=Y1
 230 LET T=T+1
 240 PRINT AT 3,16;T
 250 GOTO 50
 260 IF A<>0 OR B<>0 THEN GOTO 2
90
 270 PRINT "SIE HABEN ES GESCHAF
FT - STARK"
 280 GOTO 350
 290 PRINT "PECH GEHABT ▀ U. ▄ S
OLLEN ▚ SEIN"
 300 GOTO 350
 310 PRINT "SIE SIND MIT ▚ KOLLI
DIERT"
 350 PRINT "NOCH EINMAL ?"
 360 INPUT C$
 370 IF C$<>"N" THEN RUN
 400 STOP
 430 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒▒   ▒▒▒▒▒▒▒▒▒▒
     ▒A    ▒   ▒▒▒▒▒▒▒▒+3  ▒▒▒TI
ME▒▒  ▒▒▒▒▒▒▒▒▒  ▒▒▒-2▒▒▒▒+7▀▒
   ▒  ▒B    ▒▒  ▒▒▒▀+6▒▒▒▒▒▒  ▒▒▒▒▒
```

```
                     +7
                     -1
            +2
            +1  -7     +1    -1
               +3      -6    +1
                                 +1
                                 +1
            -1
   +3       +1              -1
    -1                   +1  -3
                        -3       +6
                                 +3
          END
          END "
 440 LET A$="-207-1-70101-106030
30102-1"
 450 LET B$="06-10103-6-3-30307-
1010101"
 460 DIM S(13,2)
 470 FOR I=1 TO 13
 480 LET S(I,1)=VAL A$(I*2-1 TO
I*2)
 490 LET S(I,2)=VAL B$(I*2-1 TO
I*2)
 500 NEXT I
 510 DIM D$(4)
 520 RETURN
 600 SAVE "CAVERNS"
 700 RUN
```

Hurkles Labyrinth

In diesem Spiel von Paul Toland durchsuchen Sie ein beliebig aufgebautes Labyrinth, mit dem Ziel, den Hurkle zu finden. Das Spiel rollt in einem realen Zeitablauf ab. Sie wandern mit Hilfe der Tasten "5", "6", "7" und "8" durch das Labyrinth und versuchen, auf das Feld zu gelangen, auf dem der Hurkle auf Sie wartet. Natürlich können Sie den Hurkle nicht sehen, da er nur unter Dilitium-Bestrahlung sichtbar wird, aber die Position des Hurkle wird Ihnen durch Nord-, Süd-, Ost- und West-Signalements auf der Anzeige oben links am Schirm angegeben. Wenn Sie den Hurkle gefunden haben, drücken Sie "N", um das Programm zu beenden, oder irgend eine andere Taste, um das Spiel zu wiederholen.
Zeile 5 baut das Labyrinth so auf, daß es in allen seinen Teilen zugänglich ist. Die Figuren in M$ haben Codes, die brauchbare Bildschirme aufbauen, wenn Sie eine RAND N-Order bekommen. Es sind viele solcher Schirme denkbar, wir haben für dieses Programm aber nur die 14 besten vorgesehen. Der Aufbau des Labyrinths läuft im SLOW-Verfahren ab. Das braucht seine Zeit, aber der Vorgang ist faszinierend zu beobachten.

```
    2 RAND
    5 LET M$="P ▚ ?>/49EHIKLN"
    7 RAND CODE M$(INT (RND*14)+1
)
   10 FOR I=1 TO 22
   20 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒"
   30 NEXT I
   40 FOR I=1 TO 30
```

```
  50 LET X=INT (RND*28)+1
  60 LET Y=INT (RND*9)*2+1
  70 LET L=INT (RND*(29-X))+2
  80 FOR J=0 TO L
  90 PRINT AT Y,X+J;" "
 100 NEXT J
 110 LET X=INT (RND*15)*2+1
 120 LET Y=INT (RND*17)+1
 130 LET L=INT (RND*(19-Y))+2
 140 FOR J=0 TO L
 150 PRINT AT Y+J,X;" "
 160 NEXT J
 170 NEXT I
 175 LET T=0
 180 LET P=PEEK 16396+256*PEEK 1
6397+1
 185 RAND
 190 LET X=INT (RND*704)+1
 195 IF PEEK (P+X)>0 THEN GOTO 1
90
 200 LET H=INT (RND*704)+1
 205 IF PEEK (P+H)>0 THEN GOTO 2
00
 210 POKE P+X,28
 220 LET D=X+(INKEY$="8")-(INKEY
$="5")+(INKEY$="6")*33-(INKEY$="
7")*33
 230 POKE P+X,0
 240 LET T=T+1
 250 IF PEEK (P+D)=0 THEN LET X=
D
 255 IF X=H THEN GOTO 350
 260 LET XD=INT (X/33)
 265 LET XA=X-XD*33
 270 LET HD=INT (H/33)
 275 LET HA=H-HD*33
 277 PRINT AT 0,5;"█████████████
█"
 280 IF XD=HD THEN GOTO 310
 290 IF HD<XD THEN PRINT AT 0,5;
"NORD"
 300 IF HD>XD THEN PRINT AT 0,5;
```

```
"SUED"
 310 IF XA=HA THEN GOTO 210
 320 IF HA<XA THEN PRINT AT 0,10
;"WEST"
 330 IF HA>XA THEN PRINT AT 0,10
;"OST "
 340 GOTO 210
 350 PRINT "SIE HABEN HURKLE IN
";T;"                  SEKUNDEN GEF
UNDEN"
 355 PRINT "NOCH EINMAL (J/N)"
 360 POKE P+H,45
 370 IF INKEY$="" THEN GOTO 370
 380 IF INKEY$="N" THEN GOTO 410
 390 CLS
 400 RUN
 410 PRINT "AUF WIEDERSEHEN"
 420 GOTO 410
 600 SAVE "HERKULES"
 700 RUN
```

Hinrichtung

In dieser witzigen, von Paul Toland erdachten Version des "Hangman"-Spiels, geben Sie 3Ø Worte in den Speicher Ihres ZX 81. Aus diesem Wort-Vorrat wählt sich der Computer 1 Wort aus. Innerhalb einer begrenzten Zeitspanne müssen Sie herausfinden, welches Wort es ist.
Beim ersten RUN des Programms werden Sie aufgefordert (Zeile 3), Ihre Worte nacheinander einzugeben. Es erhöht den Reiz des Spiels, wenn dieses Einspeichern ein anderer für Sie besorgt, sodaß Sie gar nicht wissen, welche Worte eingegeben werden. Als weitere Verfeinerung des Spiels raten wir Ihnen an, Worte zu nehmen, die eine gewisse Gemeinsamkeit haben, wie zum Beispiel „mein/dein/sein", „müde/rüde" oder „Freund/Feind". Wenn Sie die Worte eingegeben haben, können sie zusammen mit dem Programm auf einem Band SAVED werden.
Das Programm wird SAVED mit dem Befehl GOTO 39Ø. Damit stellen Sie sicher, daß das Programm beim Reloading automatisch gestartet und Ihre Worte vor einem Löschen geschützt sind, wenn Sie einen RUN-Befehl geben. Um das Programm zu starten, nachdem Sie es eingespeichert haben, drücken Sie GOTO 5.

```
 1 DIM D$(30,10)
 2 FOR I=1 TO 30
 3 INPUT D$(I)
 4 NEXT I
 5 PRINT AT 2,8;"HINRICHTUNG"
 7 PRINT AT 0,0;
10 RAND
20 LET G=0
30 LET C=G
```

```
  40 FOR I=1 TO 10
  50 PRINT I;TAB 2;"█    █"
  60 NEXT I
  70 PRINT "      ██████ "
  80 PRINT AT 8,3;"O O"
  90 PRINT AT 9,3;">=<"
  95 PRINT AT 19,3;"IHR VERSUCH"
 100 LET R=INT (RND*30)+1
 110 LET W$=""
 120 FOR I=1 TO 10
 130 IF D$(R,I)>" " THEN LET W$=
W$+D$(R,I)
 140 NEXT I
 150 LET L=LEN W$
 160 PRINT AT 8,20;"DAS WORT"
 165 LET C$="??????????"( TO L)
 170 PRINT AT 10,20;C$
 180 PRINT AT G,3;"▒▒▒▒▒"
 190 IF INKEY$="" THEN GOTO 190
 195 LET G$=INKEY$
 200 FOR I=1 TO L
 210 IF G$<>W$(I) THEN GOTO 240
 220 LET G=G-1*(G>=0)
 230 LET C$(I)=W$(I)
 240 NEXT I
 250 IF C$=W$ THEN GOTO 320
 260 PRINT AT G,3;"     "
 270 LET G=G+1
 280 IF G<10 THEN GOTO 170
 290 PRINT AT 12,3;"+ +"
 300 PRINT "     -"
 310 PRINT AT 2,20;"GEKOEPFT"
 315 GOTO 330
 320 PRINT AT 2,20;"FERTIG"
 330 PRINT AT 10,20;W$
 340 PRINT "NOCH EINMAL (J/N) ?"
 350 INPUT A$
 360 IF A$="N" THEN STOP
 370 CLS
 380 GOTO 5
 390 SAVE "EXE█"
 400 GOTO 5
```

Anagramme

Ähnlich wie bei „Hinrichtung" geben Sie bei diesem Paul Toland-Spiel 30 Worte in den Speicher Ihres Computers. Der Computer wählt eines dieser Worte aus. Sie haben dann eine begrenzte Anzahl von Versuchen, um herauszufinden, welches Wort sich der Computer ausgesucht hat.
Beim RUN des Programms werden Sie aufgefordert, 30 Worte einzugeben. Nach jedem einzelnen Wort müssen Sie ENTER drücken. Das Programm ist nun fertig. Wenn Sie es ablaufen lassen, verrät Ihnen der Computer das von ihm erwählte Wort, aber als Anagram, das heißt in einer völlig durcheinandergewürfelten Buchstabenfolge. An Ihnen ist es nun, herauszufinden, welches Wort der Computer meint.
Wenn Sie den Wort-Vorrat eingegeben haben, können Sie das Programm mit GOTO 340 aufnehmen. Damit stellen Sie sicher, daß das Programm beim Reloading automatisch läuft. Wenn Sie das Programm stoppen, nachdem Sie es eingegeben haben, und es laufen lassen wollen, ohne es zu einer Löschung der Worte kommen zu lassen, können Sie es mit GOTO 10 starten.

```
  1 DIM D$(30,10)
  2 FOR I=1 TO 30
  3 INPUT D$(I)
  4 NEXT I
 10 RAND
 20 PRINT "                    ANAGRAMM"
 30 LET R=INT (RND*30)+1
 40 LET W$=""
 50 FOR I=1 TO 10
```

```
  60 IF D$(R,I)>" " THEN LET W$=
W$+D$(R,I)
  70 NEXT I
  80 LET L=LEN W$
  90 LET S$="          "( TO L)
 100 FOR I=1 TO L
 110 LET R=INT (RND*L)+1
 120 IF S$(R)>" " THEN GOTO 110
 130 LET S$(R)=W$(I)
 140 NEXT I
 150 PRINT "IHR ANAGRAMM : ";S$
 160 FOR Y=1 TO 9
 170 PRINT Y;"  ";
 180 INPUT G$
 190 PRINT G$
 200 IF G$=W$ THEN GOTO 280
 205 PRINT "    ";
 210 FOR I=1 TO LEN G$
 215 IF I>L THEN GOTO 245
 220 IF G$(I)=W$(I) THEN PRINT "
-";
 230 IF G$(I)<>W$(I) THEN PRINT
"  ";
 240 NEXT I
 245 PRINT
 250 NEXT Y
 260 PRINT "ZEIT UM- DAS WORT WA
R ";W$
 270 GOTO 290
 280 PRINT "RICHTIG"
 290 PRINT "NOCH EINMAL (J/N) ?"
 300 INPUT A$
 310 IF A$="N" THEN STOP
 320 CLS
 330 GOTO 10
 340 SAVE "ANAM"
 350 GOTO 10
```

Reaktion

Dieser Reaktions-Test von Alistair Miller ist in BASIC und in Maschinencode geschrieben. Das Programm testet Ihre Reaktionen auf verschiedenen Geschicklichkeits-Ebenen. Beim ersten Durchlauf zeigt es im Demonstrations-Modus diese verschiedenen Geschicklichkeits-Ebenen. Der Maschinencode beansprucht 77 Bytes und ist im REM in Zeile 1 gespeichert.
Wenn das Programm läuft, geben Sie Ihre Geschicklichkeits-Ebene ein. Nach einer kurzen Pause sehen Sie, wie sich eine Linie mit der spezifizierten Geschwindigkeit über den Bildschirm bewegt. Sie müssen nun ENTER/L/K/L oder H drücken, um die Linie zu stoppen. Der Computer zeigt Ihnen an, um wieviele Einheiten sich die Linie bewegt hat, und gibt Ihnen die Chance, mit einem neuen Versuch zu einem besseren Resultat zu kommen. Sollten Sie versuchen, eine Taste zu drücken, noch ehe die Linie aufgetaucht ist, macht Ihnen der Computer klar, daß Sie gemogelt haben: Sie müssen dann das Spiel noch einmal von vorne beginnen.
Wir gehen davon aus, daß Sie keinen Assembler haben, und geben Ihnen gleich die nötigen Modifikationen für das Programm: Geben Sie das Wort REM ein, gefolgt von 73 Punkten; jetzt geben Sie die Zeilen 9000 bis 9070 ein; lassen Sie die Routine laufen; dann geben Sie das Untenstehende ein, wobei "/" als ENTER steht. (Beachten Sie dabei bitte, daß Sie in der Praxis keine Zwischenräume lassen müssen; diese sind hier nur eingerückt, damit Sie einen besseren Überblick haben; weiterhin müssen Sie darauf achten, daß Sie keine "O" eingeben, sondern Ø,

also Nullen.)
2A 0C 40 / 1E 16 / 01 21 00 / 09 09 / 09 09 / 09 09 / 09 09 / 09 09 / 22 84 40 / ED 4B 82 40 / 36 80 / 23 / 22 84 40 / 06 64 / 0E 64 / C 5 / CD BB 02 / 7D / C1 / FE BF / C8 / 0D / C2 BF 40 / 05 / C2 B4 40 / C3 9E 40 / S /

```
   1 REM U R?BBE£RND72-" 5
      6"RND GOSUB ?RND" GOSUB ?R
ND1COS E"RND0 76"RND "?:?VAL LN
 "?AT RETURN  COS $TAB  RND TAB
RND? RND
 300 PRINT AT 0,8;"REAKTIONSTEST
"
 310 PRINT "EINE ZEILE BEWEGT SI
CH UEBER DENBILDSCHIRM..STOPPEN
SIE SIE SO  SCHNELL ALS MOEGLICH
"
 320 PRINT "BENUTZEN SIE DAZU TA
STEN- ...                       "
 350 PRINT AT 11,0;"
                 "
 360 PRINT "  3   6   9  12 15 18
21"
 370 PRINT AT 21,0;"DRUECKEN SIE
 NEWLINE FUER START"
 380 FOR A=100 TO 1 STEP -5
 382 PRINT AT 21,0;"DRUECKEN SIE
 NEWLINE FUER START"
 385 POKE 16563,A
 387 POKE 16565,A
 389 PRINT AT 17,1;"SCHWIERIGKEI
TSGRAD: ";A;"   "
 390 RAND USR 16520
 400 IF INKEY$<>"" THEN GOTO 430
 410 PRINT AT 10,0;"
                    "
 420 NEXT A
 425 GOTO 380
 430 PRINT AT 10,0;"
```

```
"
 440 PRINT AT 21,0;"SCHWIERIGKEI
TSGRAD 1-100              "
 450 INPUT S
 460 IF S<1 OR S>100 THEN GOTO 4
50
 470 POKE 16563,S
 480 POKE 16565,S
 490 POKE 16514,0
 500 POKE 16515,0
 505 PRINT AT 17,1;"SCHWIERIGKEI
TSGRAD = ";S;"    "
 510 PRINT AT 14,0;"ZUFAELLIG AU
SGEWAEHLTE PAUSE"
 520 FOR A=1 TO INT (RND*200)
 530 IF INKEY$<>"" THEN GOTO 620
 540 NEXT A
 550 RAND USR 16520
 560 LET K=PEEK 16514+256*PEEK 1
6515
 570 IF K>=21 THEN GOTO 600
 580 PRINT AT 14,0;"SIE HABEN BE
I ";K;" GESTOPPT          "
 590 GOTO 660
 600 PRINT AT 14,0;"NICHT GESCHA
FFT                      "
 610 GOTO 660
 620 FOR A=1 TO 20
 630 PRINT AT 10,0;"           "
 640 PRINT AT 10,0;"BETRUEGER"
 650 NEXT A
 660 FOR A=1 TO 100
 662 IF INKEY$<>"" THEN GOTO 670
 665 NEXT A
 670 PRINT AT 10,0;"
            "
 680 PRINT AT 14,0;"
                "
 690 GOTO 380
8000 STOP
9100 SAVE "REAK"
9200 RUN
```

Lebenswandels-Bilanz

Dieses Programm zieht die Bilanz der Veränderungen, die sich in letzter Zeit in Ihren Lebensumständen ergeben haben, und erklärt Ihnen die Auswirkungen dieser Veränderungen auf ihr Ego. Jede Veränderung in den Lebensumständen, egal ob positiv oder negativ, erzeugt einen Stress. Auch Weihnachten schafft Stress, gar nicht zu reden von einem neuen Job oder Familienzuwachs. Der Bildschirm konfrontiert Sie mit einer ganzen Reihe von Möglichkeiten, zum Beispiel wirft er die Möglichkeit einer SCHWANGERSCHAFT auf. Wenn Sie also schwanger sind, drücken Sie "Y" (für YES/JA). Sind Sie nicht schwanger oder gehören gar dem männlichen Geschlecht an, so drücken Sie "N". Zum hoffentlich guten Ende faßt der Computer Ihre Lage zusammen und erteilt Ihnen eine Lektion über Ihren Lebenswandel.

```
  150 PRINT "           [illegible]
[illegible]"
  155 PRINT
  160 PRINT "  WIE VERAENDERUNGEN
  IN IHREM       LEBENSWANDEL IHRE
  ZUKUNFT              BEEINFLUSSE
N"
  170 PRINT AT 20,1;"ANTWORTEN SI
E MIT JA ODER NEIN"
  180 LET L=0
  190 GOTO 250
  200 PRINT AT 10,0;L$;"
                                 "
  210 IF INKEY$<>"J" AND INKEY$<>
"N" THEN GOTO 210
  220 IF INKEY$="J" THEN LET L=L+
VAL S$(I*2-1 TO I*2)
```

```
  230  LET  I=I+1
  232  FOR  K=1  TO  10
  234  NEXT  K
  240  RETURN
  250  LET  I=1
  260  LET  S$="99737165636353504745454440403939383736353331302929292826262524232120202019191817161515131211"
  270  LET  L$="PARTNER  GESTORBEN"
  275  GOSUB  200
  280  LET  L$="GESCHIEDEN"
  285  GOSUB  200
  290  LET  L$="HARTE  DROGEN  AUFGEGEBEN"
  295  GOSUB  200
  300  LET  L$="GETRENNTE  EHE"
  305  GOSUB  200
  310  LET  L$="GEFAENGNISSTRAFE"
  315  GOSUB  200
  320  LET  L$="TOT  EINES  FAMILIENMITGLIEDS"
  325  GOSUB  200
  330  LET  L$="VERLETZT  ODER  KRANKHEIT"
  335  GOSUB  200
  340  LET  L$="HEIRAT"
  345  GOSUB  200
  350  LET  L$="ENTLASSEN  WORDEN"
  355  GOSUB  200
  360  LET  L$="VERSOEHNUNG  IN  DER  EHE"
  365  GOSUB  200
  370  LET  L$="PENSIONIERT  WORDEN"
  375  GOSUB  200
  380  LET  L$="IST  EIN  FAMILIENMITGLIED  KRANK"
  385  GOSUB  200
  390  LET  L$="SIND  SIE  SCHWANGER"
  395  GOSUB  200
  400  LET  L$="HABEN  SIE  IHRE  40  ZIGARETTEN  PROTAG  AUFGEGEBEN"
```

```
 405 GOSUB 200
 410 LET L$="HABEN SIE IHRE LAST
 BEIM SEX"
 415 GOSUB 200
 420 LET L$="ERWARTEN SIE FAMILI
ENZUWACHS"
 425 GOSUB 200
 430 LET L$="IHRE GESCHAEFTE NEU
 GEORDNET"
 435 GOSUB 200
 440 LET L$="FINANZIELLE VERAEND
ERUNGEN"
 445 GOSUB 200
 450 LET L$="IST EIN FREUND GEST
ORBEN"
 455 GOSUB 200
 460 LET L$="HABEN SIE EINE ANDE
RE ARBEIT"
 465 GOSUB 200
 470 LET L$="STREITEN SIE OFT MI
T DEM PARTNER"
 475 GOSUB 200
 480 LET L$="MENSTRUATIONSBESCHW
ERDEN"
 485 GOSUB 200
 490 LET L$="KREDITE ODER PFAEND
UNGEN"
 495 GOSUB 200
 500 LET L$="SCHULDEN UEBER 2000
0 DM"
 505 GOSUB 200
 510 LET L$="NEUE VERANTWORTUNG
IN DER ARBEIT"
 515 GOSUB 200
 520 LET L$="FLUGVERSPAETUNG"
 525 GOSUB 200
 530 LET L$="HAT EINES DER KINDE
R DAS HAUS    VERLASSEN"
 535 GOSUB 200
 540 LET L$="HABEN SIE AERGER MI
T SCHWAGERN"
 545 GOSUB 200
```

```
 550 LET L$="BESONDERE LEISTUNG
VOLLBRACHT"
 555 GOSUB 200
 560 LET L$="HAT DER PARTNER BEG
ONNEN ODER    AUFGEHOERT ZU ARBEI
TEN"
 565 GOSUB 200
 570 LET L$="HABEN IHRE KINDER M
IT DER SCHULEAUFGEHOERT ODER ANG
EFANGEN"
 575 GOSUB 200
 580 LET L$="HABEN SICH IHRE LEB
ENSUMSTAENDE GEAENDERT"
 585 GOSUB 200
 590 LET L$="VERAENDERUNGEN IHRE
R GEWOHNHEIT"
 595 GOSUB 200
 600 LET L$="AERGER MIT DEM CHEF
"
 605 GOSUB 200
 610 LET L$="HABEN SIE AUFGEHOER
T MEHR ALS 40ZIGARETTEN AM TAG Z
U RAUCHEN"
 615 GOSUB 200
 620 LET L$="HABEN SICH IHRE ARB
EITSSTUNDEN  ODER GEWOHNHEITEN G
EAENDERT"
 625 GOSUB 200
 630 LET L$="HATTEN SIE EINEN WO
HNUNGSWECHSEL"
 635 GOSUB 200
 640 LET L$="HATTEN SIE EINEN SC
HULWECHSEL"
 645 GOSUB 200
 650 LET L$="ERHOLEN SIE SICH SO
 VIEL WIE    FRUEHER"
 655 GOSUB 200
 660 LET L$="BETREIBEN SIE NOCH
RELIGIOESE    AKTIVITAETEN"
 665 GOSUB 200
 670 LET L$="HAT SICH IHR GESELL
SCHAFTLICHES LEBEN GEAENDERT"
```

```
 675 GOSUB 200
 680 LET L$="HYPOTHEKEN ODER SCH
ULDEN UNTER  20000 DM"
 682 GOSUB 200
 684 LET L$="HABEN SIE DIE SCHLA
FGEWOHNHEITENGEAENDERT"
 686 GOSUB 200
 690 LET L$="IST DIE ZAHL DER FA
MILIENTREFFENGESUNKEN ODER GESTI
EGEN"
 695 GOSUB 200
 700 LET L$="HABEN SIE IHRE KULI
NARISCHEN     GEWOHNHEITEN GEAEND
ERT"
 705 GOSUB 200
 710 LET L$="NAEHERN SICH DIE FE
RIEN"
 715 GOSUB 200
 720 LET L$="STEHT WEIHNACHTEN V
OR DER TUER"
 725 GOSUB 200
 730 LET L$="HABEN SIE DAS GESET
Z UEBERTRETEN"
 735 GOSUB 200
 740 PRINT AT 7,1;"IHRE L.E.W. T
ESTPUNKTZAHL IST"
 741 PRINT AT 8,13;L
 750 PRINT ,,"            DAS URTEI
L-         "
 760 IF L<150 THEN PRINT "SIE FU
EHREN EINEN SICHEREN,      STABIL
EN LEBENSWANDEL UND IHNEN WIRD K
AUM EIN UNFALL ODER EINE  KRANKH
EIT ZUSTOSSEN."
 765 IF L=0 THEN PRINT "---BIS S
IE AN LANGEWEILE STERBEN"
 770 IF L>149 AND L<200 THEN PRI
NT "ES GIBT EINE 37 PROZENTIGE
    WAHRSCHEINLICHKEIT, DASS IHN
EN  EIN GESUNDHEITLICHES ODER FI
NAN-ZIELLES PROBLEM BEGEGNET."
 780 IF L>199 AND L<300 THEN PRI
```

```
NT "ES GIBT EINE 51 PROZENTIGE
    WAHRSCHEINLICHKEIT, DASS SIE
    GESUNDHEITLICHEN ODER FINANZ
I-  ELLEN SCHWIERIGKETEN BEGEGNE
N ; SIE SOLLTEN IHREN LEBENSWAND
EL  ORDNEN."
 790 IF L>299 THEN PRINT "** PAS
SEN SIE AUF ** BEI IHNEN  LIEGT
EINE 79 PROZENTIGE WAHR-  SCHEIN
LICHKEIT, DASS SIE GESUND-HEITLI
CHEN ODER FINANZIELLEN    SCHWIE
RIGKEITEN ENTGEGENSEGELN. GEBEN
SIE DIE HOFFNUNG NICHT AUFPASSEN
 SIE NUR AUF SICH AUF UND FUEHRE
N EIN RUHIGES LEBEN, BIS  IHRE P
UNKTZAHL SINKT."
 800 PRINT ,,"NOCH MEHR KLIENTEN
 ?"
 810 INPUT A$
 820 IF A$="N" THEN STOP
 830 CLS
 840 GOTO 1
 900 SAVE "LEBE█"
 901 RUN
```

Armada

Dies ist ein sehr raffiniert ausgearbeitetes Seekriegsspiel von Jim Archer. Es basiert auf dem bekannten Spiel „Schiffe versenken“, wartet aber mit etlichen Erweiterungen und Variationen auf, die es im Original-Spiel nicht gibt. Die Kriegsflotten bestehen hier aus sechs verschiedenen Schiffstypen: Sechs Schlachtschiffe, fünf Kreuzer, vier Torpedoboote, drei Flugzeugträger, zwei U-Boote und ein Zerstörer. Merke: Je weniger Schiffe in die Schlacht geworfen werden, desto höher der Punktgewinn beim Versenken eines Schiffes!
Während der eine der beiden Spieler ein nützliches Buch liest oder sonstwie seine Augen diskret abwendet, gibt der andere die Positionen seiner Schiffe ein. Der Computer hat in diesem Spiel nur die Funktion eines Schiedsrichters. Es ist wichtig, die Positionen der einzelnen Schiffe zu überprüfen: jedes Schiff muß nahe bei den anderen der gleichen Seite sein, mit anderen Worten: die Schiffe der beiden Seiten müssen geschlossene Formationen bilden.
Einem der beiden Spieler wird vom Computer das Recht zur Eröffnung des Feuers erteilt, und dann beschießen beide Gegner abwechselnd das Gebiet des anderen. Wer einen Treffer landet, darf noch eine Salve feuern. Jedes mit einem Schuß belegte Feld wird sofort weiß, damit Sie wissen, daß es sinnlose Vergeudung von Munition und Zeit wäre, noch einmal darauf zu feuern. Wenn Sie es trotzdem nicht lassen können, gibt Ihnen der Computer den Hinweis SCHON VERSUCHT.
Sollten Sie das Programm zerstören (was eigentlich

kaum möglich ist, weil der Computer unlautere Vorschläge zurückweist), können Sie es mit GOTO 2500 wieder aufbauen.

```
  10 DIM A$(4,10,10)
  15 PRINT AT 0,13;"ARMADA"
  16 PAUSE 50
  20 DIM O$(2,3)
  30 DIM S(2)
  35 PRINT
  40 PRINT "NAME DES 1.SPIELERS
(3 BUCHST.)?"
  47 INPUT O$(1)
  48 PRINT O$(1)
  50 PRINT "NAME DES 2.SPIELERS
(3 BUCHST.)?"
  55 INPUT O$(2)
  57 PRINT O$(2)
  60 LET M=0
  70 LET T=6
  80 LET W=1
  90 LET W$="  "
  95 LET G$="123456789"
  97 FAST
 100 GOSUB 900
 105 SLOW
 110 LET H=0
 120 GOSUB 1100
 130 FOR X=1 TO 2
 140 LET S(X)=0
 155 GOSUB 350
 160 FOR R=T TO 1 STEP -1
 200 FOR U=1 TO R
 205 LET H=H+7-R
 207 PRINT AT 21,0;R;"  ";B$(R,4
TO 3+VAL (B$(R,2 TO 3)));
 209 PRINT " :                   ";
 210 PRINT AT 21,21;"NO.";U;"?  "
;AT 21,28;"    ";AT 21,28;
 220 GOSUB 2000
 225 GOTO 1700
```

```
 230 LET A$(X,I,J)=B$(R,4)
 235 PRINT AT 2*J,5+2*I;B$(R,4)
 240 NEXT J
 270 NEXT R
 280 NEXT X
 300 LET W=3
 310 LET W$="▒"
 315 FAST
 320 GOSUB 900
 325 SLOW
 330 GOTO 700
 350 CLS
 351 PRINT O$(X);" BITTE IHRE FL
OTTENPOSITION"
 352 PAUSE 100
 353 CLS
 355 FAST
 360 PRINT O$(X);"     A B C D E
F G H I J"
 365 PRINT
 370 FOR Z=1 TO 10
 380 IF Z=10 THEN GOTO 1000
 390 PRINT "       ";G$(Z);
 400 FOR Y=1 TO 10
 410 IF A$(X,Y,Z)<>" " THEN PRIN
T TAB (2*Y+1);A$(X,Y,Z);
 420 NEXT Y
 430 PRINT
 440 PRINT
 450 NEXT Z
 470 SLOW
 480 RETURN
 500 CLS
 510 PRINT
 520 PRINT
 530 PRINT "      ABCDEFGHIJ    AB
CDEFGHIJ"
 540 FOR Z=1 TO 10
 550 FOR X=3 TO 4
 555 IF Z=10 THEN GOTO 1020
 560 PRINT "    ";G$(Z);
 570 FOR Y=1 TO 10
```

```
 580 PRINT A$(X,Y,Z);
 585 NEXT Y
 600 NEXT X
 605 PRINT
 610 NEXT Z
 620 PRINT
 630 PRINT
 640 PRINT TAB 5;O$(1);TAB 22;O$
(2)
 650 PRINT
 660 PRINT TAB 4;"PUNKTE ";S(1);
TAB 21;"PUNKTE ";S(2)
 665 PRINT AT 19,5;"SCHIESSEN :"
 670 RETURN
 700 GOSUB 500
 710 LET K=INT (RND*2+1)
 720 LET M=1
 740 LET P=3-K
 745 PRINT AT 21,0;"
                     "
 750 PRINT AT 19,0;O$(K);AT 19,1
5;"    ";AT 19,15;
 760 GOSUB 2000
 770 IF A$(P,I,J)<>" " THEN GOTO
 1500
 780 PRINT AT 21,0;"WASSER"
 785 LET A$(P,I,J)="."
 786 LET A$(P+2,I,J)=" "
 787 PRINT AT J+2,3+14*(P-1)+I;"
"
 790 PAUSE 80
 800 LET K=P
 810 GOTO 740
 900 FOR X=W TO W+1
 910 FOR Y=1 TO 10
 920 FOR Z=1 TO 10
 930 LET A$(X,Y,Z)=W$
 940 NEXT Z
 950 NEXT Y
 960 NEXT X
 970 RETURN
1000 PRINT "      [illegible]";
```

```
1010 GOTO 400
1020 PRINT "  ▓▓▓";
1030 GOTO 570
1060 PRINT AT 21,0;"NICHT ERLAUB
T                   "
1070 PAUSE 50
1090 GOTO 207
1100 DIM B$(6,18)
1110 LET B$(1)="110ZERSTOERER"
1120 LET B$(2)="206U-BOOT"
1130 LET B$(3)="315FLUGZEUGTRAEG
ER"
1140 LET B$(4)="411TORPEDOBOOT"
1150 LET B$(5)="507KREUZER"
1160 LET B$(6)="614SCHLACHTSCHIF
F"
1170 RETURN
1200 IF M=1 THEN GOTO 1300
1210 PRINT AT 21,0;"WIE BITTE ?
                  ";AT 21,19;
1220 GOTO 2000
1300 PRINT AT 21,0;"WIE BITTE ?"
1310 PAUSE 50
1320 PRINT AT 19,15;"   ";AT 21,
0;"                     ";AT 19,
15;
1330 GOTO 2000
1450 PRINT AT 21,0;"SCHON VERSUC
HT";
1460 IF A$(P,I,J)<>"." THEN PRIN
T "-RUINEN UEBRIG";
1470 PAUSE 50
1475 PRINT AT 21,0;"
                "

1480 GOTO 750
1500 IF A$(P,I,J)="*" OR A$(P,I,
J)="." THEN GOTO 1450
1510 LET A$(P+2,I,J)=A$(P,I,J)
1515 PRINT AT J+2,3+14*(P-1)+I;A
$(P,I,J)
1520 LET V=1
1530 IF B$(V,4)=A$(P,I,J) THEN G
```

```
OTO 1560
1540 LET V=V+1
1550 GOTO 1530
1560 PRINT AT 21,0;B$(V,4 TO 18)
;" GETROFFEN";
1570 PRINT " ZP ";7-V
1575 PAUSE 120
1580 LET S(K)=S(K)+7-V
1582 LET A$(P,I,J)="*"
1583 PRINT AT 17,11+(K-1)*17;S(K
)
1585 IF S(K)=H/2 THEN GOTO 1600
1590 GOTO 743
1600 CLS
1610 PRINT
1620 PRINT "GANZE FLOTTE ZERSTOE
RT"
1630 PRINT
1640 LET V=S(K)-S(P)
1650 PRINT O$(K);" SIEGT MIT ";V
;" PUNKTEN"
1660 STOP
1700 IF A$(X,I,J)<>" " OR I<1 OR
 I>10 OR J<1 OR J>10 THEN GOTO 1
060
1710 LET G=0
1720 LET C=I-1
1730 LET E=J-1
1740 IF I=1 THEN LET C=1
1750 IF J=1 THEN LET E=1
1760 LET D=I+1
1770 LET F=J+1
1780 IF I=10 THEN LET D=10
1790 IF J=10 THEN LET F=10
1800 IF A$(X,C,E)=" " THEN GOTO
1830
1810 IF A$(X,C,E)<>B$(R,4) THEN
GOTO 1060
1820 LET G=1
1830 IF C<D THEN GOTO 1920
1840 IF E<F THEN GOTO 1960
1850 IF U>1 AND G=0 THEN GOTO 10
```

```
60
1860 GOTO 230
1920 LET C=C+1
1930 GOTO 1800
1960 LET E=E+1
1970 LET C=I-1
1980 IF I=1 THEN LET C=1
1990 GOTO 1800
2000 INPUT F$
2010 PRINT F$
2015 IF LEN F$<2 OR LEN F$>3 THE
N GOTO 1200
2017 IF CODE F$(1)<38 OR CODE F$
(1)>47 OR CODE F$(2)<29 OR CODE 
F$(2)>37 THEN GOTO 1200
2020 LET I=CODE F$(1)-37
2030 IF LEN F$=3 THEN GOTO 2060
2040 LET J=VAL F$(2)
2050 RETURN
2060 IF F$(3)<>"0" THEN GOTO 120
0
2065 LET J=VAL F$(2 TO 3)
2070 GOTO 2050
2500 GOSUB 500
2510 GOTO 740
3000 SAVE "U-BOO█"
3010 RUN
```

Golf für zwei

Jim Archer hat dieses Programm ursprünglich als einfaches Golfspiel entworfen, aber dann zu einem Spiel für zwei erweitert, den Golfplatz mit allerhand Raffinessen ausgebaut und sogar die Rolle von Wind-Richtung und -Geschwindigkeit beim Golfen berücksichtigt. Ein Zufallsfaktor von $\mp$15 Prozent sorgt dafür, daß das Spiel menschlich bleibt.
Der ZX 81 entscheidet, wer den ersten Schlag führen darf, dann schlagen beide Spieler abwechselnd mit angemessener Kraft. Die Kraft, mit der ein Schlag ausgeführt wird, hängt natürlich von der Entfernung zum Loch ab, das getroffen werden soll. Diese Distanz wird in Yards gemessen. Auf dem Hauptfeld entspricht 1 Quadrat 1Ø Yards, auf der Grünfläche entspricht 1 Quadrat 1 Yard. Es kann ein Winkel zwischen −18Ø Grad und +18Ø Grad gewählt werden. Die Position des Balles nach ausgeführtem Schlag wird unter Berücksichtigung des Windeffekts und des Zufallsfaktors errechnet und als Initial des Spielers auf dem nächstgelegenen Quadrat ausgedruckt.
Landet ein Spieler auf einem der Bunker, so hat er eine 3:4-Chance, die Situation zu klären. Landet der Ball auf der Grünfläche, so wird dort ein beliebiges Quadrat als neue Position bestimmt. Wenn Sie einen Treffer landen, wird Ihnen eindrucksvoll vorgeführt, wie der Ball in das Loch fällt. Am Ende des 8-Loch-Kurses wird der Spieler bewertet, was sein Selbstbewußtsein wahrscheinlich derart stärkt, daß er sich sofort in ein neues Spiel stürzt.

```
   15  DIM N$(2,3)
   20  PRINT "         ***GOLF FUER Z
WEI***"
   21  PRINT
   22  PRINT "           EIN 8 LOCH S
PIEL"
   23  PRINT
   24  PRINT "NAME DES 1. SPIELERS
  (3 BUCHST.)"
   25  INPUT N$(1)
   26  PRINT
   28  PRINT N$(1)
   30  PRINT "NAME DES 2. SPIELERS
  (3 BUCHST.)"
   31  PRINT
   32  INPUT N$(2)
   33  PRINT N$(2)
   38  PRINT ,,"WINDGESCHWINDIGKEI
T ? ";
   40  INPUT W
   50  PRINT W
   60  IF W<0 THEN GOTO 1000
   70  IF W>60 THEN GOTO 1025
   75  LET W1=PI/180*W
   80  PRINT "WINDRICHTUNG ? ";
   90  INPUT E$
   95  PRINT E$
  100  LET D$=E$
  110  LET V=-1
  120  GOSUB 1100
  130  IF V=-1 THEN GOSUB 1040
  140  IF V=-1 THEN GOTO 80
  141  DIM C(2)
  142  DIM O(2)
  143  DIM T(2)
  144  DIM I(2)
  147  DIM J(2)
  148  DIM B(2)
  150  LET U=V+180
  160  IF U>=360 THEN LET U=U-360
  170  RAND
  180  GOSUB 1200
```

```
 190 LET N=0
 200 LET C(1)=1
 201 LET C(2)=1
 202 DIM E(2)
 203 LET N=N+1
 204 DIM H(2)
 205 LET H=0
 210 LET A=VAL A$(5*(N-1)+3 TO 5
*(N-1)+5)
 220 LET D$=A$(5*(N-1)+1 TO 5*(N
-1)+2)
 230 IF D$(2)=" " THEN LET D$=D$
(1)
 240 GOSUB 1300
 250 CLS
 260 PRINT "[illegible] ";N;" [illegible] ";F
$;"         ";AT 0,26;A;" M"
 265 GOSUB 1870
 270 DIM G$(20,32)
 280 GOSUB 1100
 285 LET PO=V*4/45
 290 LET I=VAL E$(PO+1 TO PO+2)
 295 LET J=VAL E$(PO+3 TO PO+4)
 302 LET M=INT (RND*2)+1
 310 LET G$(I,J)=N$(M,1)
 320 PRINT AT I,J;N$(M,1)
 325 LET V1=V*PI/180
 330 LET X=I-INT (A/10*COS V1+.5
)
 340 LET Y=J+INT (A/10*SIN V1+.5
)
 345 LET G$(X,Y)="▒"
 350 PRINT AT X,Y;"▒"
 360 FOR R=X-1 TO X+1
 370 FOR S=Y-1 TO Y+1
 380 IF NOT (R=X AND S=Y) THEN G
OSUB 1400
 390 NEXT S
 400 NEXT R
 402 LET P=3-M
 406 LET I(1)=I
 407 LET I(2)=I
```

```
 408 LET J(1)=J
 409 LET J(2)=J
 410 GOSUB 1700
 411 IF C(M)>1 THEN PRINT AT 0,2
3;"SCHLAG ";C(M);" "
 412 GOSUB 415
 414 GOTO 510
 415 PRINT AT 21,0;N$(M);" SCHLA
GSTAERKE ?       ";AT 21,26;
 420 INPUT K
 430 PRINT K
 440 PAUSE 50
 450 PRINT AT 21,0;"MIT EINEM WI
NKEL VON       ";AT 21,21;
 460 INPUT T
 470 PRINT T;" GRAD"
 475 LET T(M)=T(M)+1
 476 PRINT AT M+1,30;T(M)
 480 IF E(M)=1 THEN GOSUB 1830
 490 LET K=K+.4*(RND-.5)*K
 500 LET T=T+30*(RND-.5)
 505 RETURN
 510 LET Q=.004*K*W
 515 IF E(M)=1 THEN GOTO 585
 520 PRINT AT I(M),J(M);" "
 525 LET G$(I(M),J(M))=" "
 528 IF C(P)=1 THEN PRINT AT I,J
;N$(P,1)
 550 LET T1=T*PI/180
 560 LET I(M)=I(M)-INT ((K*COS T
1+Q*COS W1)/10)
 570 LET J(M)=J(M)+INT ((K*SIN T
1+Q*SIN W1)/10)
 575 IF I(M)=I(P) AND J(M)=J(P)
THEN LET J(M)=J(M)+1
 580 GOSUB 1430
 585 IF G$(I(M),J(M))="█" THEN G
OSUB 1800
 590 PRINT AT I(M),J(M);N$(M,1)
 595 LET G$(I(M),J(M))=N$(M,1)
 605 LET C(M)=C(M)+1
 610 IF ABS (X-I(M))<2 AND ABS (
```

```
Y-J(M))<2 THEN GOSUB 2200
 611 IF C(P)>0 THEN GOSUB 1260
 612 IF H(M)=1 THEN GOTO 620
 614 GOTO 411
 620 CLS
 630 PRINT "LOCH ";N;" :AUF DEM
GRUEN:"
 635 FAST
 636 GOSUB 1870
 637 IF H=1 THEN GOTO 2495
 640 DIM H$(9,9)
 645 LET H=1
 650 FOR R=1 TO 9
 660 FOR S=1 TO 9
 675 LET H$(R,S)="▒"
 680 IF R=5 AND S=5 THEN LET H$(
R,S)="█"
 690 PRINT AT R+2,S+9;H$(R,S)
 700 NEXT S
 710 NEXT R
 742 SLOW
 745 LET H$(I(M),J(M))=N$(M,1)
 750 PRINT AT I(M)+2,J(M)+9;N$(M
,1)
 755 IF H(P)=1 AND C(P)>0 THEN P
RINT AT I(P)+2,J(P)+9;N$(P,1)
 760 IF I(M)=5 AND J(M)=5 THEN G
OTO 1590
 765 PRINT AT 0,24;"SCHLAG ";C(M
)
 770 GOSUB 415
 780 PRINT AT I(M)+2,J(M)+9;"▒"
 790 LET H$(I(M),J(M))="▒"
 800 LET T1=T*PI/180
 810 LET I(M)=I(M)-INT (K*COS T1
+.5)
 820 LET J(M)=J(M)+INT (K*SIN T1
+.5)
 830 IF H(M)=H(P) THEN GOSUB 186
0
 840 GOSUB 1480
 845 IF C(P)>0 AND I(M)=I(P) AND
```

```
J(M)=J(P) THEN LET J(M)=J(M)+1
 850 PRINT AT I(M)+2,J(M)+9;N$(M
,1)
 860 IF I(M)=5 AND J(M)=5 THEN G
OTO 1600
 870 LET H$(I(M),J(M))=N$(M,1)
 880 PRINT AT I(M)+2,J(M)+9;N$(M
,1)
 900 LET C(M)=C(M)+1
 910 IF C(P)=0 THEN GOTO 765
 920 GOSUB 1260
 950 IF H(M)=1 THEN GOTO 765
 980 GOTO 1900
1010 PRINT "SEIEN SIE ETWAS MUTI
GER"
1020 GOTO 30
1025 PRINT "ZU STARKER WIND----
SPIELAUSFALL"
1030 STOP
1040 PRINT "WIE BITTE ?"
1050 RETURN
1100 IF D$="N" THEN LET V=0
1110 IF D$="NO" THEN LET V=45
1120 IF D$="O" THEN LET V=90
1130 IF D$="SO" THEN LET V=135
1140 IF D$="S" THEN LET V=180
1150 IF D$="SW" THEN LET V=225
1160 IF D$="W" THEN LET V=270
1170 IF D$="NW" THEN LET V=315
1180 RETURN
1200 LET A$="NW240W 280N 1755023
0N 1805025ØNO2200 275"
1210 LET E$="2015200311010103011
50128113020028"
1220 LET B$="0413052111251627161
41301090606060"
1250 RETURN
1260 LET M=P
1270 LET P=3-M
1280 RETURN
1300 IF D$="N" THEN LET F$="NORD
"
```

```
1310 IF D$="NO" THEN LET F$="NOR
N-OST"
1320 IF D$="O" THEN LET F$="OST"
1330 IF D$="SO" THEN LET F$="SUE
D-OST"
1335 IF D$="S" THEN LET F$="SUED
"
1340 IF D$="SW" THEN LET F$="SUE
D-WEST"
1350 IF D$="W" THEN LET F$="WEST
"
1360 IF D$="NW" THEN LET F$="NOR
D-WEST"
1370 RETURN
1400 LET G$(R,S)="▒"
1410 PRINT AT R,S;"▒"
1420 RETURN
1430 IF I(M)<1 THEN LET I(M)=1
1440 IF I(M)>20 THEN LET I(M)=20
1450 IF J(M)<1 THEN LET J(M)=1
1460 IF J(M)>31 THEN LET J(M)=31
1470 RETURN
1480 IF I(M)<1 THEN LET I(M)=1
1490 IF I(M)>9 THEN LET I(M)=9
1500 IF J(M)<1 THEN LET J(M)=1
1510 IF J(M)>9 THEN LET J(M)=9
1520 RETURN
1590 LET C(M)=C(M)-1
1600 PRINT AT 7,14;"*"
1601 LET G$(X,Y)=N$(M,1)
1620 PRINT AT 21,0;"         ▓▓▓ EI
NGELOCHT ▓▓▓         "
1625 PAUSE 140
1630 LET C(M)=0
1635 PRINT AT 7,14;"▓"
1640 IF C(P)>0 THEN GOTO 2400
1660 IF N<8 THEN GOTO 200
1665 FOR M=1 TO 2
1670 CLS
1675 LET TOT=T(M)
1680 PRINT AT 7,10;N$(M);AT 9,6;
"**********";AT 10,6;"GESAMT ";T
```

```
(M);AT 11,6;"***********";AT 15,0
;"URTEIL  ";
1681 IF TOT<=20 THEN PRINT "A -
SEHR GUT"
1682 IF TOT>20 AND TOT<=25 THEN
PRINT "B - GUT"
1683 IF TOT>40 THEN PRINT "F - [illegible]
[illegible]"
1685 IF TOT>25 AND TOT<=30 THEN
PRINT "C - DURCHSCHNITT"
1686 IF TOT>30 AND TOT<=35 THEN
PRINT "D - SCHLAPP"
1687 IF TOT>35 AND TOT<=40 THEN
PRINT "E - WOHL ANFAENGER"
1688 STOP
1689 NEXT M
1690 STOP
1700 LET A=VAL B$(PO+1 TO PO+2)
1710 LET B=VAL B$(PO+3 TO PO+4)
1720 IF D$="N" OR D$="S" THEN GO
TO 1780
1730 FOR D=A TO A+2
1740 LET G$(D,B)="█"
1750 PRINT AT D,B;"█"
1760 NEXT D
1770 RETURN
1780 LET G$(A,B TO B+2)="███"
1790 PRINT AT A,B;"███"
1795 RETURN
1800 PRINT AT 21,0;"        [illegible] IM
 BUNKER [illegible]          "
1805 PAUSE 90
1810 LET B(M)=1
1820 RETURN
1830 IF RND<.25 THEN RETURN
1840 LET B(M)=0
1850 RETURN
1870 PRINT AT 1,27;"[illegible]";AT 2,26
;N$(1);":";T(1);AT 3,26;N$(2);":
";T(2)
1880 RETURN
1900 FAST
```

```
1905 CLS
1906 PRINT "LOCH ";N;" DISTANZ "
;F$
1907 GOSUB 1870
1910 FOR R=X-1 TO X+1
1920 FOR S=Y-1 TO Y+1
1930 PRINT AT R,S;"▒"
1940 NEXT S
1950 NEXT R
1952 PRINT AT X,Y;"█"
1955 PRINT AT I(M),J(M);N$(M,1)
1960 SLOW
1970 GOTO 410
2000 DIM H(2)
2010 PRINT H(1),H(2)
2020 LET H(1)=1
2030 PRINT H(1),H(2)
2200 LET I(M)=4+3*(I(M)-X)+INT (
RND*10/3.34)
2210 LET J(M)=4+3*(J(M)-Y)+INT (
RND*10/3.34)
2220 LET H(M)=1
2225 IF H(P)=1 AND I(M)=I(P) AND
 J(M)=J(P) THEN LET J(M)=J(M)+1
2230 RETURN
2400 GOSUB 1260
2410 IF H(M)=0 THEN GOTO 1900
2420 GOTO 760
2496 CLS
2497 PRINT "LOCH ";N;" : AUF DEM
 GRUEN : "
2498 GOSUB 1870
2505 FAST
2510 FOR R=1 TO 9
2520 FOR S=1 TO 9
2530 IF R=5 AND S=5 THEN LET H$(
R,S)="█"
2540 PRINT AT R+2,S+9;H$(R,S)
2550 NEXT S
2560 NEXT R
2570 GOTO 742
9000 SAVE "GOLF"
9010 RUN
```

Der ZX 81-Berater

Dieses hervorragende Programm von Chris Callender ist die ideale Lebenshilfe für jedermann, der sich in heiklen Situationen nicht entscheiden kann. Es führt in vielen Fällen zu einer ausgeglichenen Beurteilung der Lage und kann deshalb wirklich sehr nützlich werden. Bei dem nachstehend dargestellten Beispiel geht es um die Anschaffung eines Haustieres; dieses Beispiel soll Ihnen klarmachen, wie Sie Ihre eigenen Problemfälle per ZX 81 entscheidungsreif machen können.

Anzahl der Möglichkeiten? 3
Anzahl der wichtigen Faktoren? 3
1. Möglichkeit? Schwein
2. Möglichkeit? Schwan
3. Möglichkeit? Hamster
Berater II
1. Faktor? Kosten
2. Faktor? Zuneigung
3. Faktor? Farbe
Berater II
Wie veranschlagen Sie Kosten für ein Schwein? 55 etc.
Danke ... Hier ist meine Meinung
Schwein 326
Schwan 293
Hamster 162

```
  30 GOSUB 8000
  40 PRINT "ZAHL DER MOEGLICHKEI
TEN: ";
  50 GOSUB 9000
  60 LET O=VAL B$
  70 PRINT "ZAHL DER WICHTIGEN F
AKTOREN: ";
  80 GOSUB 9000
  90 LET F=VAL B$
 100 DIM O$(O,32)
 110 DIM F$(F,32)
 120 CLS
 130 GOSUB 8000
 140 FOR A=1 TO O
 145 IF PEEK 16442<4 THEN SCROLL

 150 PRINT A;". MOEGLICHKEIT"
 155 IF PEEK 16442<4 THEN SCROLL

 160 GOSUB 9000
 170 LET O$(A)=B$
 180 NEXT A
 190 CLS
 200 GOSUB 8000
 210 FOR A=1 TO F
 220 IF PEEK 16442<4 THEN SCROLL

 230 PRINT A;". FAKTOR ?"
 240 IF PEEK 16442<4 THEN SCROLL

 250 GOSUB 9000
 255 LET F$(A)=B$
 260 NEXT A
 270 DIM M(O,F)
 275 CLS
 277 GOSUB 8000
 280 FOR A=1 TO O
 290 FOR B=1 TO F
 295 CLS
 297 GOSUB 8000
 300 PRINT "WIE HOCH SCHAETZEN S
IE"
```

```
 301 PRINT F$(B)
 310 PRINT "FUER ";O$(A)
 320 GOSUB 9000
 330 LET M(A,B)=VAL B$
 340 NEXT B
 350 NEXT A
 360 CLS
 370 GOSUB 8000
 380 PRINT "DANKE.....HIER IST M
EINE MEINUNG"
 390 FOR A=1 TO O
 400 LET C=0
 410 FOR B=1 TO F
 420 LET C=C+M(A,B)
 430 NEXT B
 440 PRINT O$(A);C
 450 NEXT A
 460 PRINT "NOCH EIN DURCHGANG ?
 ";
 470 GOSUB 9000
 480 CLS
 490 IF B$(1)="J" THEN RUN
 500 STOP
8000 PRINT TAB 5;"     BERATER II
    "
8010 PRINT TAB 5;"██████████████
███"
8020 RETURN
9010 LET B$=""
9020 PAUSE 4E4
9022 LET A$=INKEY$
9025 IF A$=CHR$ 118 THEN GOTO 90
60
9027 IF INKEY$=CHR$ 115 THEN LET
 A$=" "
9030 PRINT A$;
9040 LET B$=B$+A$
9050 GOTO 9020
9060 PRINT
9070 RETURN
9100 SAVE "BERATER"
9200 RUN
```

Audio Morse Test

Dieses Programm von Chris Callender wählt sich ein Wort und spielt es in Morse-Code durch den Lautsprecher Ihres Fernsehers. Die entsprechenden Geräusche werden über Maschinencode erzeugt.
Das erste Programm baut den Maschinencode in einem REM-Statement in Zeile 10 auf. Wenn das erste Programm gelaufen ist, drücken Sie LIST; Sie werden nun die Zeichen sehen, nachdem der REM geändert wurde. Wenn Sie diese Methode in ihren eigenen Programmen verwenden wollen, geben Sie das erste Programm ein und löschen dann die Zeilen 20 bis 50. Zur Erzeugung des Tons geben Sie folgendes ein:

```
POKE 16523, duration
LET L USR 16514
```

Um zu dem Morse-Test zu gelangen, geben Sie das zweite Programm ein. Wenn Sie statt der 30 in Zeile 390 die Zahl der Worte eingeben, die Sie benutzen wollen, können Sie nach Zeile 1350 noch mehr Worte einfügen.
Der Maschinencode ist in Zeile 10 gespeichert. Hier ein Beispiel für das Disassembling:

```
LD A,00
EXX
CALL 0320
EXX
INC A
CP duration
JP nz,4084
RET
```

```
  10 REM Y ▒LN 4▀▒W RETURN ▒TAB
▜RNDTAN
  20 DIM M$(36,5)
  30 LET M$(1)="-----"
  40 LET M$(2)=".----"
  50 LET M$(3)="..---"
  60 LET M$(4)="...--"
  70 LET M$(5)="....-"
  80 LET M$(6)="....."
  90 LET M$(7)="-...."
 100 LET M$(8)="--..."
 110 LET M$(9)="---.."
 120 LET M$(10)="----."
 130 LET M$(11)=".-"
 140 LET M$(12)="-..."
 150 LET M$(13)="-.-."
 160 LET M$(14)="-.."
 170 LET M$(15)="."
 180 LET M$(16)="..-."
 190 LET M$(17)="--."
 200 LET M$(18)="...."
 210 LET M$(19)=".."
 220 LET M$(20)=".---"
 230 LET M$(21)="-.-"
 240 LET M$(22)=".-.."
 250 LET M$(23)="--"
 260 LET M$(24)="-."
 270 LET M$(25)="---"
 280 LET M$(26)=".--."
 290 LET M$(27)="--.-"
 300 LET M$(28)=".-."
 310 LET M$(29)="..."
 320 LET M$(30)="-"
 330 LET M$(31)="..-"
 340 LET M$(32)="...-"
 350 LET M$(33)=".--"
 360 LET M$(34)="-..-"
 370 LET M$(35)="-.--"
 380 LET M$(36)="--.."
 390 LET N=INT (RND*30)
 400 GOSUB 740+N*20
 410 FOR M=1 TO LEN A$
```

```
 420  LET  C=CODE  A$(M)-27
 430  FOR  I=1  TO  5
 440  IF  M$(C,I)="."  THEN  POKE  16
523,10
 450  IF  M$(C,I)="-"  THEN  POKE  16
523,40
 460  IF  M$(C,I)=" "  THEN  GOTO  51
0
 470  LET  L=USR  16514
 480  FOR  A=1  TO  10
 490  NEXT  A
 500  NEXT  I
 510  FOR  A=1  TO  50
 520  NEXT  A
 530  NEXT  M
 540  PRINT  "WIE  HEISST  DAS  WORT
?"
 550  INPUT  B$
 560  IF  B$=A$  THEN  GOTO  680
 570  PRINT  "FALSCH"
 580  PRINT  "NOCH  EINMAL  VORSPIEL
EN  (J/N)  ?"
 590  PRINT  "UM  DIE  LOESUNG  ZU  ER
FAHREN        DRUECKEN  SIE  DIE  [illegible]  T
ASTE"
 600  INPUT  B$
 610  CLS
 620  IF  B$="N"  THEN  GOTO  540
 630  IF  B$="J"  THEN  GOTO  410
 640  IF  B$="F"  THEN  PRINT  A$
 650  IF  B$="F"  THEN  STOP
 660  PRINT  "FALSCHE  ANTWORT.....
"
 670  GOTO  560
 680  CLS
 690  PRINT  AT  10,8;"RICHTIG.....
."
 700  PAUSE  50
 710  PRINT  AT  10,8;"[illegible]
[illegible]"
 720  PAUSE  50
 730  GOTO  630
```

```
 740 LET A$="SOS"
 745 RETURN
 760 LET A$="SINCLAIR"
 770 RETURN
 780 LET A$="ZX"
 790 RETURN
 800 LET A$="COMPUTER"
 810 RETURN
 820 LET A$="FREUND"
 830 RETURN
 840 LET A$="FEIND"
 850 RETURN
 860 LET A$="HALLO"
 870 RETURN
 880 LET A$="GEHIRN"
 890 RETURN
 900 LET A$="TIER"
 910 RETURN
 920 LET A$="RS232"
 930 RETURN
 940 LET A$="CHRIS"
 950 RETURN
 960 LET A$="MORSE"
 970 RETURN
 980 LET A$="SEX"
 990 RETURN
1000 LET A$="HORROR"
1010 RETURN
1020 LET A$="MICRO"
1030 RETURN
1040 LET A$="PRINTER"
1050 RETURN
1060 LET A$="REGENBOGEN"
1070 RETURN
1080 LET A$="REICHTUM"
1090 RETURN
1100 LET A$="ZX99"
1110 RETURN
1120 LET A$="CASSETTE"
1130 RETURN
1140 LET A$="16KRAM"
1150 RETURN
```

```
1160 LET A$="GLAS"
1170 RETURN
1180 LET A$="AUTO"
1190 RETURN
1200 LET A$="HILFE"
1210 RETURN
1220 LET A$="FUELLER"
1230 RETURN
1240 LET A$="KOENIG"
1250 RETURN
1260 LET A$="DAME"
1270 RETURN
1280 LET A$="RADIO"
1290 RETURN
1300 LET A$="TUER"
1310 RETURN
1320 LET A$="SCHLUESSEL"
1330 RETURN
1340 LET A$="GUT"
1350 RETURN
9000 SAVE "MORSE"
9010 RUN
```

Der ZX 81-Charmeur

Wenn Sie die ganze Nacht vergeblich an einem Programm gearbeitet haben, Ihre Augen schmerzen, Ihr Kühlschrank leer und Ihre Freundin abgehauen ist, dann können Sie sich nur noch mit diesem wunderbaren 1K-Programm von Dilwyn Jones wieder aufbauen. Sie brauchen nur eine beliebige Taste zu drücken, und der Computer verwöhnt Sie mit einer verbalen Streicheleinheit. Das Programm läßt sich natürlich auch benutzen, um Beleidigungen (siehe nächstes Programm), Koseworte oder klangvolle Figuren-Namen für Ihr nächstes Abenteuer-Programm zu kreieren. Die Graphik-Elemente in B$ kriegen Sie durch GRAPHICS Shift 1, Shift 5, Shift D und die letzte Graphik durch Shift INKEY$. Die Graphiken in D$ kommen von Shift 1 und Shift E.
Alle in diesem Programm benutzten Worte werden in den beiden Strings A$ und C$ verankert. Die Strings B$ und D$ beinhalten Tips, deren CODE dort steht, wo jedes Wort in A$ beziehungsweise C$ beginnt. A ist die Variable, die festlegt, welches Wort aus A$ genommen wird. Entsprechend ist C die Variable für die Auswahl der Worte aus C$. Die Zeile 8Ø zieht das Display sauber scrollend nach oben, sodaß jedes Wort unter dem voraufgegangenen erscheint. Das ganze Dekodieren und Ausdrucken passiert in Zeile 9Ø, und Zeile 1ØØ sorgt dafür, daß die Komplimente unaufhörlich weiterfließen.

```
   10 RAND
   20 LET A$="NETTERWEISERSCHOENE
RSCHLAUERCHARMANTERWITZIGERFREUN
DLICHERGROSSER"
   30 LET B$="▀ ▛$+1BJUPI"
   40 LET C$="MANNPROGRAMMIERERCH
AMEURGAUNERINDIVIDUALISTHELDLIEB
LINGSCHNURZEL"
   50 LET D$="▀█ >;3GKSINKEY$"
   60 LET A=INT (RND*8)+1
   70 LET B=INT (RND*8)+1
   80 SCROLL
   90 PRINT A$(CODE B$(A) TO CODE
 B$(A+1)-1);" ";C$(CODE D$(B) TO
 CODE D$(B+1)-1)
  100 GOTO 60
  900 SAVE "CHAMEUR"
  910 RUN
```

Der ZX 81-Krawall-Generator

Dieses Program ist eine bösartige Variante des "ZX 81-Charmeur" und stammt ebenfalls von Dilwyn Jones. Es sorgt dafür, daß einem nicht die Worte ausgehen, wenn man seinen Nächsten beleidigen und erniedrigen will – ist also ebenso geeignet für häusliche Auseinandersetzungen wie für Büro-Krawalle. Die Graphik-Elemente in BS sind Shift 1, Shift T und Shift S, die entsprechenden Elemente in DS sind Shift 1, Shift 5 und Shift S.

```
   10 RAND
   20 LET A$="VOLLGEFRESSENERLAEC
HERLICHERDUMMERIDIOTISCHERBLOEDE
RVERDREHTER"
   30 LET B$="▀ (17IPZ"
   40 LET C$="ARSCHUHUIDIOTHORNOC
HSESACKHUNDFETTWANSTSUFFKOPPSCHE
ISSERKAKKER"
   50 LET D$="▀▄▀▒:*.3CKTZ"
   60 LET A=INT (RND*6)+1
   70 LET B=INT (RND*10)+1
   80 SCROLL
   90 PRINT A$(CODE B$(A) TO CODE
 B$(A+1)-1);" ";C$(CODE D$(B) TO
 CODE D$(B+1)-1)
   98 FOR I=1 TO 20
   99 NEXT I
  100 GOTO 60
  900 SAVE "KRAWAL█"
  910 RUN
```

Magisches Dreieck à la Loubere

Nick Wilson hat sich dieses Programm ausgedacht, das nach der Methode de la Loubere vier magische Quadrate von verschiedener Größe erzeugt. Ein magisches Quadrat ist ein Viereck, in dem alle Zahlen der senkrechten und waagrechten Reihen und der Diagonalen die gleiche Summe ergeben. Beim Anlaufen des Programms werden Sie aufgefordert, die Seite/Höhe des Quadrats zu bestimmen; Sie dürfen dabei nur ungerade Zahlen zwischen 1 und 9 angeben, was von Zeile 130 überprüft wird. Dann wird der Bildschirm blank und das Quadrat wird aufgebaut, um anschließend im Printer COPYed zu werden.

```
  100 DIM M(47,47)
  110 PRINT AT 10,0;"ANZAHL DER K
OLUMNEN/REIHEN ? "
  120 INPUT N
  125 CLS
  130 IF N/2=INT (N/2) THEN GOTO
110
  131 PRINT AT 0,0;"      MAGISCHE
 NUMMER ";AT 1,0;N;"      REIHEN/K
OLUMNEN"
  150 LET C1=0
  160 LET C=INT (N/2)+1
  170 LET R=1
  180 LET C1=C1+1
  190 LET M(R,C)=C1
  191 PRINT AT R*2+1,C*3;C1
  200 IF C1=INT (N**2) THEN GOTO
330
  210 IF C1/N<>INT (C1/N) THEN GO
```

```
TO 240
  220 LET R=R+1
  230 GOTO 180
  240 LET C=C+1
  250 IF C<=N THEN GOTO 290
  260 LET C=1
  270 LET R=R-1
  280 GOTO 180
  290 LET R=R-1
  300 IF R>0 THEN GOTO 180
  310 LET R=N
  320 GOTO 180
  330 LET T=0
  350 FOR I=1 TO N
  360 LET T=T+M(I,1)
  370 NEXT I
  380 PRINT AT 0,0;T
  390 COPY
  900 SAVE "MAGISC█"
  910 RUN
```

```
15      MAGISCHE NUMMER
3       REIHEN/KOLUMNEN

    8    1    6

    3    5    7

    4    9    2
```

Wie schreibt man bessere Programme?

von Tim Hartnell, Herausgeber der Reihe

In diesem Buch gibt es eine Reihe sehr guter Programme, und in vielen Computerzeitschriften findet man andere dieser Art. Aber gleichgültig, wie gut veröffentlichte Programme sein mögen, Sie haben mehr Spaß dabei, sie zu fahren, wenn sie teilweise oder ganz von Ihnen selbst verfaßt sind. Programme mit Ihrem eigenen Stempel zu versehen, sie nach Ihren Wünschen und Einfällen abzuändern, ist eine ausgezeichnete Methode, die Programme zu verbessern, und mit der Zeit werden Sie natürlich ein besserer und vielseitigerer Programmierer. Programme in Zeitschriften und in Bänden wie diesen sind als Ausgangspunkte für Ihre eigenen Entwicklungen ideal. Sie werden vielleicht auch feststellen, daß neu angekündigte Software-Pakete fruchtbare Anregungen liefern können. Sie brauchen nur die Beschreibung dessen zu lesen, was das auf den Markt gebrachte Programm leistet, und tun damit den ersten Schritt auf dem Weg zu Ihrem eigenen Programm. Sie müssen natürlich darauf achten, nicht gegen das Copyright, also die Rechte, auf das Bildschirmdisplay, den Namen des Programms oder die Namen der „Figuren" im Programm zu verstoßen. Sie werden aber wohl feststellen, daß das Programm ab einem bestimmten Punkt in seiner Entwicklung ein Eigenleben annimmt, vom Ansatzpunkt aus weiterwächst und sich verwandelt, bis Sie am Ende ein Programm mit völlig neuer Konzeption und Struktur haben.

Was Sie auch tun, hüten Sie sich davor, anderer Leute Arbeit als Ihre eigene auszugeben. Ändern und verbessern Sie veröffentlichte Programme nach Herzenslust, reichen Sie sie dann aber nicht bei Zeitschriften ein, als wären es Originalprogramme. Ich kann nicht mehr zählen, wie oft eines meiner Programme aus einem meiner Bücher mir zur Veröffentlichung vorgelegt worden ist.
Halten Sie stets Ausschau nach neuen Ideen, wenn Sie Bücher, Spiele- und Computerzeitschriften durchblättern oder sich in Video-Spielhallen umsehen. Es mag sich lohnen, Ideen zu Spielen, Figurenformen, Geräuschen, dramatischen Spielabschlüssen und dergleichen zu notieren. So wird es Ihnen an Einfällen nie fehlen, und Sie werden außerdem das Material einbauen können, um bessere Spiele hervorzubringen, die das Interesse des Spielers länger fesseln.
Spiele fallen in der Regel in drei Kategorien, und es lohnt sich, über die Kategorie zu vergewissern, in die Ihr vorgesehenes Programm fallen wird, **bevor** Sie mit dem Programmieren anfangen, weil die Spielkategorie auf die Programmierweise einen erheblichen Einfluß hat. Das soll nicht heißen, daß ein Programm, an dem Sie arbeiten, nicht von einer Kategorie in eine andere rutschen oder daß ein bestimmtes Spiel nicht zwei Kategorien umgreifen kann, aber es ist trotzdem nützlich, die verschiedenen Gruppen bei Ihren Überlegungen getrennt zu halten, damit Sie klarer denken. Die drei Kategorien sind:

1. Brettspiele
2. „Arkade"-Spiele (das heißt schnelle, laute Echtzeitspiele, bei denen sich für das Auge viel tut)
3. Glücksspiele (wie Roulette und Snap)

Bei Brettspielen ist die Qualität des Spiels wichtiger als blitzschnelle Reaktion, während die Arkade-Programme um jeden Preis in Bewegung gehalten werden müssen, selbst wenn das bedeutet, einen Teil „Intelligenz" Ihrer Angreifer vom Mars zu opfern. Glücksspiele hängen mehr von der Leichtigkeit des Spielens („benutzerfreundliche" Eingaben) und einer Annäherung an echte Zufälligkeit ab als die beiden anderen Kategorien.
Sie werden feststellen, daß Spielprogramme sich nach Typen unterscheiden, die Unterabteilungen der oben erwähnten Kategorien sind. Viele Brettspiele sind Abwandlungen von Schach oder Dame; viele Arkade-Spiele fingen an als Spiele vom Typ „Angreifer aus dem Weltall"; und Glücksspiele stammen aus der „wirklichen" Welt der Würfel und Spielkarten. Wenn Sie sich eine Programmbeschreibung oder ein Spielgerät ansehen und das Spiel einzustufen versuchen, können Sie mithelfen, neue Ideen zu entwickeln, die in den Gesamtbereich dieses Spiels passen.
Beim Programmieren gibt es eine Denkweise – allgemein „strukturiertes Programmieren" genannt – nach deren Ansicht zu Beginn des Verfahrens, Spiele zu schreiben, sei Disziplin unabdingbar. Das macht zwar weniger Spaß, als sich gleich an den Computer zu setzen, aber dafür kommt am Ende auch ein besseres Programm heraus. Ich habe einmal ein Programm namens "Dome Dweller" (Raumstation) geschrieben, ein Simulationsprogramm, in dem der Spieler für eine „Wohnkuppel auf dem Mond" zu sorgen hat und entscheiden muß, welche Erzeugnisse hergestellt und verkauft werden müssen, damit man für die Bewohner der Station Sauerstoff und Nahrungsmittel erwerben kann. Als ich mir über den all-

gemeinen Ablauf im klaren war, arbeitete ich das Bildschirmdisplay aus und kam auf folgende Idee:
Die Sauerstoffwerte sind gering;
Im Jahr 3 leben in Ihrer Kuppel 96 Menschen;
An Geld sind vorhanden 5693 Dollar;
Die jährliche Wartung kostet 226 Dollar;
Die Sauerstofftanks enthalten 811 Einheiten;
Sauerstoff kostet je Einheit 8 Dollar;
Jeder Kuppelbewohner verbraucht im Jahr 5 Einheiten;
Die Lebensmittelvorräte stehen bei 2122;
Jeder Kuppelbewohner verbraucht im Jahr 3 Einheiten (jeder 6 Dollar, für die ganze Kuppel 576 Dollar. Das reicht bei der jetzigen Bewohnerzahl für 7 Jahre). Sie können Ihre konkurrenzlosen Mondskulpturen im Tauschhandel den Leuten in anderen Kuppeln überlassen. Jedes Stück erfordert 2 Einheiten Sauerstoff, verkauft wird es für 30 Dollar.

Wie Sie sich nach dieser „Protokollprobe" denken können, kommt es bei dem Programm auf die Entscheidung an, wie viele Mondskulpturen Sie herstellen und verkaufen müssen, um Sauerstoff und Nahrungsmittel zu erwerben und die „jährlichen Wartungskosten" zu bezahlen. Das Problem ist, daß die Herstellung jeder Skulptur Sauerstoff erfordert, so daß Sie Ihren Wunsch, Geld zu verdienen, in Übereinstimmung mit der Notwendigkeit bringen müssen, den Sauerstoff klug einzuteilen.
Sie haben vielleicht Lust, ein solches Programm selbst zu schreiben. Am Ende müßte es viel Spaß machen, und Sie können Ihre Programmierkünste dadurch sicherlich erweitern. Als erstes stellt man eine Liste darüber auf, was das Programm leisten soll:

Die erforderlichen Variablen festlegen;
Dem Spieler den „Zustand der Kuppel" mitteilen;
Fragen, wieviel Sauerstoff gekauft werden soll;
Prüfen, ob man sich das leisten kann, wenn ja, kaufen, sonst zurückgehen und erneut fragen;
Fragen, wieviel an Nahrungsmitteln gekauft werden soll;
Prüfen, ob man sich das leisten kann, wenn ja, kaufen, sonst zurückgehen und erneut fragen;
Menge des Sauerstoffs aktualisieren;
Menge der Nahrungsmittel aktualisieren;
Geld auf Restsumme verringern;
Fragen, wie viele Skulpturen hergestellt werden sollen;
Prüfen, ob genug Sauerstoff vorhanden ist, um so viele herzustellen, sonst zurückgehen und erneut fragen;
Sauerstoffmenge um jenen Betrag verringern, der erforderlich ist, um die gewünschte Zahl von Skulpturen herzustellen, Geldgesamtsumme erhöhen um Wert der hergestellten Skulpturen;
Die Bewohnerzahl gering erhöhen, für das „laufende Jahr" 1 hinzufügen;
Prüfen, ob genug Nahrungsmittel auf Lager sind, um die ganze Bewohnerschaft zu ernähren;
Prüfen, ob für alle Bewohner genug Sauerstoff vorhanden ist;
Prüfen, ob Geld vorhanden ist;
Falls irgendeine dieser Bedingungen nicht erfüllt ist (etwa nicht genug Nahrungsmittel), Spiel beenden;
Wenn alle erfüllt sind, in der Schleife zurückgehen, dem Spieler Zustand der Kuppel mitteilen und weitermachen.

Sie könnten vermutlich ein Programm „Raumstation" mit Hilfe der obigen Liste und der Information in der „Protokollprobe" schreiben. Es gibt jedoch ein Geheimnis, das Programmierprobleme beinahe augenblicklich aufklärt, und das ich mit Ihnen teilen möchte. Sie können alle wichtigen Teile eines Programms tatsächlich innerhalb von Minuten schreiben, sehen das Rohgerüst eines solchen Programms also schon laufen, lange bevor Sie sich mit den Einzelheiten abgeben. Sobald Sie ein Gerüst haben, können Sie daran arbeiten, so lange Sie wollen, weil Sie wissen, daß Sie in jedem Augenblick der Programmierentwicklung ein funktionierendes Programm haben. Sie brauchen nicht bis zum Ende zu warten, um zu sehen, wie Sie vorankommen. Das „Geheimnis" besteht darin, das ganze Programm in eine Folge von Subroutinen-Aufrufen einzufügen, alles in einer Dauerschleife. Die allerersten Zeilen, die Sie in Ihren Computer eingeben, sehen demnach so aus:

```
 10 REM RAUMSTATION
 20 GOSUB 1000: REM VARIABLE ZUTEILEN
 30 GOSUB 2000: REM ZUSTAND D. KUPPEL ANZEIG.
 40 GOSUB 3000: REM SAUERSTOFF
 50 GOSUB 4000: REM NAHRUNGSMITTEL
 60 GOSUB 5000: REM SKULPTUR
 70 GOSUB 6000: REM BEWOHNERZAHL AKTUALISIER.
 80 GOSUB 7000: REM ZUSTAND D. KUPPEL PRUEFEN
 90 IF (alle Bedingungen nach GOSUB 7000 positiv)
    THEM GOTO 30
100 REM Ende des Spiels
```

Wie Sie sehen, ist es, sobald Sie auf diese Weise die Hauptschleife aufgebaut haben, relativ einfach, eine Subroutine nach der anderen auszufüllen, sie dabei zu testen und auszuarbeiten, bis Sie am Ende ein sehr gutes Programm haben. Das einzige, was Sie nun noch brauchen, ist eine Liste der Variablen, die Sie im Programm verwenden.

Ich halte es für das Beste, klare Namen für Variable zu nehmen, damit Sie beim Programmieren nicht nachsehen müssen, ob AA für die Bewohnerzahl oder für die Zahl der verbrauchten Sauerstoffeinheiten je Skulptur steht. Damit Programme zwischen verschiedenen Computern so leicht wie möglich zu übertragen sind, können Sie bei Variablennamen aus zwei Buchstaben bleiben oder (falls Ihr Computer das zuläßt) den Vorteil langer Namen für Variable nutzen (wie OXYVERBRAUCH für die Menge verbrauchten Sauerstoffs). Dann treten keine Zweifel darüber auf, was jeder Variablenname bedeutet. Um zu zeigen, daß das geht, und einen weiteren Vorteil deutlicher Variablennamen zu veranschaulichen, hier die Variablen in "Dome Dweller":

LEUTE – Bewohnerzahl der Kuppel;
GELD – Barvermögen;
NAHR – vorhandene Lebensmittelvorräte;
NAHRKOST – wieviel jede Einheit Nahungsmittel kostet;
NAHRBED – wie viele Einheiten Nahrungsmittel im Jahr pro Person verbraucht wurden;
SKULKOST – wieviel Sauerstoff für jede Skulptur verbraucht wurde;
SKULPREIS – für wieviel Dollar jede Skulptur verkauft wurde;

OXY – vorhandene Sauerstoffvorräte;
OXYBED – wie viele Einheiten Sauerstoff im Jahr pro Person verbraucht wurden;
REPAR – die jährlichen Reparaturkosten für die Kuppel;
JAHR – das Lebensjahr der Kuppel.

Klare Variablennamen auf diese Weise zu verwenden – obwohl sie mehr Speicherplatz verbrauchen als Variablennamen mit einem oder zwei Buchstaben – vereinfacht es sehr, ein Programm durchzugehen und festzustellen, was jeder Programmteil leistet. Überdies, und das ist der zweite Vorteil, den ich erwähnt habe, fällt es beim Verfassen des Programms sehr leicht, die für Berechnungen erforderlichen Formeln einzufügen. Damit meine ich: Wenn Sie beispielsweise (wie ich in diesem Programm) eine Anzeige dafür einbauen wollen, wieviel Sauerstoff jedes Jahr gebraucht wird, multiplizieren Sie einfach die Zahl der Menschen in der Kuppel (LEUTE) mit der Zahl der Sauerstoffeinheiten, die jede Person im Jahr braucht (OXYBED). Dann können Sie in die Listen für den Zustand der Kuppel Zeilen einfügen wie:

```
PRINT „IN DER KUPPEL“; „GIBT ES“; „LEUTE“
PRINT „IM JAHR“; JAHR
PRINT „JEDE PERSON BRAUCHT“; OXYBED;
  „EINHEITEN“
PRINT „SAUERSTOFF IM JAHR“; OXYBED*LEUTE;
  „ERFORDERLICH“
PRINT „FUER DIE GANZE KUPPEL“
```

Außerdem wird dadurch die Prüfung sehr erleichtert, ob Einkäufe möglich sind. Beispiel: Um Nahrungsmittel zu kaufen, würden Sie sagen:

```
PRINT „WIEVIEL NAHRUNGSMITTEL WOLLEN SIE
  KAUFEN?"
INPUT A
IF A*NAHRUNGSKOST>GELD THEN GOTO
  (neues A holen)
```

Zu den hier gemachten Vorschlägen, Ihre Programme durch Anwendung „strukturierten Programmierens“ zu verbessern, gehören:

- ein Probeexemplar oder Modell des endgültigen Bildschirmdisplays skizzieren;
- eine Liste dessen aufstellen, was das Programm bei jedem Lauf durch eine „Hauptsteuerungs-Schleife“ zu leisten hat;
- diese Liste in eine Folge von aufzurufenden Subroutinen verwandeln;
- wo möglich, klare Variablennamen verwenden.

Wenn Sie Programme für andere Leute entwerfen, ist es nützlich, ganz deutlich zu machen, was der Spieler beim Fahren des Programms tun soll. Es hat, vor allem bei begrenztem Speicherplatz, wenig Sinn, in das Programm selbst eine lange Folge von Instruktionen einzufügen, aber schriftlich niederlegen sollten Sie solche Instruktionen schon. Außerdem sollten Anforderungen für Anwender deutlich sein (etwa: EINGABE, WIE OFT SIE ES VERSUCHEN WOLLEN) und Hinweise auf die Grenzen für die Eingaben enthalten (Beispiel: MIT WIEVIEL KARTEN WOLLEN SIE ANFANGEN: 1,2 ODER 3?).
Sie dürfen nicht davon ausgehen, Sie wären jedesmal dabei, wenn ein Programm gefahren wird, also sollten Sie Ihr Bestes tun, um es so narrensicher wie möglich zu machen. Wenn Sie können, fügen Sie in das Programm

Fehleraufdeckungs-Routinen ein, um dafür zu sorgen, daß ein Fehler bei der Eingabe einer Möglichkeit unter mehreren zu einem früheren Zeitpunkt im Programm keinen Absturz oder später unsinnige Ergebnisse verursacht.

Wenn Sie diesen Abschnitt des Buches mehrmals durchgelesen haben und versuchen, diese Ideen auf Ihre eigene Programmiertätigkeit anzuwenden, sollten Sie feststellen, daß die Qualität Ihrer Leistung deutlich zunimmt, und daß Sie auch mehr Zeit für Verbesserung und Ausschmückung eines Programms aufwenden können als für die ziemlich mechanische Aufgabe, das Ding zum Laufen zu bringen.

Fachausdrücke

A

Akkumulator – der Ort im Computer, wo arithmetische Berechnungen ausgeführt und die Resultate dieser Berechnungen gespeichert werden.

Algorithmus – die Reihe von Schritten, die ein Computer durchläuft, um ein bestimmtes Problem zu lösen.

alphanumerisch – dieser Ausdruck wird in der Regel im Verhältnis zu einer Tastatur verwendet, wie bei „es handelt sich um eine alphanumerische Tastatur". Das bedeutet, daß die Tastatur über Buchstaben und Ziffern verfügt. Man gebraucht ihn auch, wenn man vom „Zeichensatz" des Computers spricht. Der Zeichensatz umfaßt die Ziffern und Buchstaben, die der Computer auf dem Bildschirm anzeigen kann.

ALU (Arithmetic/Logic Unit) – Rechen- und Steuerwerk. Der Teil des Computers, der Arithmetik (wie Addition und Subtraktion) leistet, und wo Entscheidungen getroffen werden.

AND – eine Boolesche logische Operation, die der Computer bei seinem Entscheidungsprozeß ausführt. Sie beruht auf der Booleschen Algebra, ein System, entwickelt von dem Mathematiker George Boole (1815–1864). In der Booleschen Algebra stellen die Variablen eines Ausdrucks eine logische Operation wie OR oder NOR dar.

Anweisung – Daten, die einen Einzelschritt in der Informationsverarbeitung durch den Computer steuern (auch Befehl genannt).

ASCII – für American Standard Code for Information Exchange, das gebräuchlichste Codierungssystem für

alphanumerische Zeichen der englischen Sprache. Es gibt 129 große und kleine Buchstaben, Ziffern und einige Sonderzeichen. ASCII verwandelt die Symbole und Steueranweisungen in Binärkombinationen von sieben Bits.

Assembler – ein Programm, das andere, in Assemblersprache geschriebene Programme in Maschinencode verwandelt (den der Computer direkt verstehen kann). Assemblersprache ist eine niedere Programmiersprache, die leicht zu merkende Kombinationen von zwei oder drei Buchstaben zur Darstellung eines bestimmten Befehls verwendet, den der Assembler dann umwandelt, so daß die Maschine ihn verstehen kann. Beispiele dafür sind ADD (addiere) und SUB (subtrahiere). Ein in Assemblersprache programmierter Computer vermag schneller zu arbeiten als ein in einer höheren Programmiersprache wie BASIC programmierter.

B

BASIC – Abkürzung für Beginners All-Purpose Symbolic Instruction Code. Die gebräuchlichste Computersprache für Mikrocomputer. Obwohl von vielen Leuten kritisiert, hat sie den Vorteil, leicht erlernbar zu sein. Viele BASIC-Befehle entsprechen dem Alltagsenglisch.

Baud – benannt nach Emile Baudot, dem Wegbereiter der Telegraphie. Baud ist das Maß für die Übertragungsgeschwindigkeit und entspricht rund einem Bit pro Sekunde.

BCD – Abkürzung für Binary Coded Decimal, Binärcode für Dezimalziffern.

Benchmark – Vergleichslauf, bei dem bestimmte Funk-

tionen des Computers gemessen werden können. Es gibt eine Reihe sogenannter „Standard-Benchmarkaufgaben", die in der Regel aber nur die Schnelligkeit messen, bei einem Mikrocomputer selten der Punkt, der für den künftigen Käufer das Wichtigste wäre.

Betriebssystem – die mit dem Computer in der Regel mitgelieferte Soft- oder Firmware, mit denen Sie andere Programme fahren können.

Binär – ein Zahlensystem, das nur Nullen und Einsen verwendet.

Bit – Abkürzung für Binary Digit, Binärziffer. Das ist die kleinste Einheit, die eine Computerschaltung erkennen kann.

Boolesche Algebra – das algebraische System, das der Mathematiker George Boole entwickelt hat. Es verwendet algebraische Schreibung, um logische Beziehungen auszudrücken (siehe AND).

Bug – eigentlich „Käfer", „Insekt" – Fehler in einem Computerprogramm, der das Programm hindert, richtig zu laufen. Obwohl damit in der Regel ein Fehler oder ein Ausfall in einem Programm gemeint ist, kann "bug" auch einen Defekt in der Computer-Hardware bedeuten.

Bus – Sammelschiene für die Übertragung von Signalen wie Datenanweisungen oder Leistungsein- und ausgabe eines Computers.

Byte – eine Gruppe von Binärziffern, die ein Computerwort ausmacht. In der Regel besteht ein Byte aus acht Bits.

C

CAI – Computer Assisted Instruction = computerunterstützter Unterricht.

CAL – Computer Assisted Learning = computerunterstütztes Lernen. Der Ausdruck wird allgemein für Programme verwendet, die den Lernenden mit dem Lernprozeß befaßt.

Chip – der übliche Ausdruck für die gesamte Schaltung, die auf eine kleine Siliziumscheibe geätzt wird. Der Chip befindet sich natürlich im Herzen des Computers.

COBOL – eine höhere Programmiersprache, abgekürzt für Common Business Orientated Language. COBOL ist in erster Linie für Dateiverwaltung und Archivierung gedacht.

Compiler – ein Computerprogramm, das höhere Programmiersprachen in binären Maschinencode umwandelt, damit der Computer damit umgehen kann.

Computer – ein Gerät mit drei Hauptfähigkeiten oder -funktionen:
1. Daten anzunehmen;
2. Probleme zu lösen;
3. Resultate zu liefern.

CPU – Abkürzung für Central Processing Unit = Zentraleinheit. Das Herz der Computerintelligenz, wo Daten verarbeitet und Befehle ausgeführt werden. Bei Mikros Mikroprozessor.

Cursor – ein Zeichen, das auf dem Bildschirm erscheint, wenn der Computer in Betrieb ist. Es zeigt an, wo das nächste Zeichen angezeigt wird. Ein Computer verfügt in aller Regel über „Cursor-Steuertasten", damit der Anwender den Cursor auf dem Bildschirm herumführen kann.

D

Datei – eine Sammlung von untereinander in Beziehung stehenden Informationsposten, auf systematische Weise organisiert.
Daten – Information in einer Form, die der Computer verarbeiten kann.
Debug – Debug = entfehlern, der gängige Ausdruck dafür, ein Programm durchzugehen und alle Fehler darin auszubessern, das heißt, den Bugs nachzujagen und sie zu entfernen (siehe Bug).
Digitalcomputer – ein Computer, der mit Informationen arbeitet, die in diskreter Form erscheinen.
Diskette – eine magnetisierte Plastikscheibe, etwas kleiner als eine Single-Schallplatte. Sie wird verwendet zum Speichern und Abrufen von Daten. Disketten sind erheblich schneller zu laden als eine Kassette mit dem Programm der selben Länge. Die Diskette kann sehr rasch gelesen werden, während ein Programm für zusätzliche Daten gefahren wird.
Display – die sichtbare Ausgabe eines Computers, in der Regel auf einem Fernseh- oder Monitorbildschirm.
Dynamischer Speicher – eine Speichereinheit im Computer, der seinen Inhalt vergißt, wenn der Strom abgeschaltet wird.

E

Editor – der Ausdruck wird allgemein verwendet für die Routine im Computer, mit der man beim Schreiben eines Programms Zeilen verändern kann.
EPROM – Abkürzung für Erasable Programmable

Read-Only Memory = löschbarer programmierbarer Festspeicher. Im Computer wie ROM, nur kann man in einen EPROM verhältnismäßig leicht Material laden, und es verschwindet nicht, wenn man den Strom abschaltet. EPROMs müssen, um gelöscht zu werden, in starkes UV-Licht gebracht werden.

F

Fehlermeldungen – die Information, die ein Computer übermittelt, wenn in einem Programmteil ein Fehler in der Codierung auftritt. In der Regel dadurch angezeigt, daß der Computer stehenbleibt und unten am Bildschirm ein Wort oder ein Wort und Zahlen oder nur Zahlen anzeigt. Das teilt Ihnen mit, welcher Fehler begangen worden ist. Häufige Fehler: die Verwendung des Buchstaben „O" statt der Null in einer Programmzeile, das Weglassen beider oder einer Klammer in einem Ausdruck, oder das Versäumnis, eine Variable zu definieren.
Floppy Disk – eine vergleichsweise billige Magnetscheibe für das Speichern von Computerinformation, so genannt, weil sie biegsam ist (siehe Diskette).
Flow Chart – Flow Chart = Flußdiagrammm, ein Diagramm, angelegt, bevor man ein Programm schreibt. Die Hauptoperationen werden in Rechtecke, Trapeze oder andere Gebilde gesetzt und durch Striche miteinander verbunden, wobei Pfeile Schleifen darstellen und an den Verzweigungsstellen Entscheidungen notiert werden. Das Abfassen eines Programms wird dadurch sehr erleichtert, weil folgenreiche Fehler wie Endlosschleifen oder nicht definierte Variable schon in einem frühen Stadium erkannt werden können. Bei sehr kurzen Program-

men mag sich ein Flußdiagramm nicht lohnen, aber im allgemeinen nützt es beim Abfassen vom Programmen.
Firmware – es gibt bei Computern drei Arten von "ware": Software, das sind „temporäre" Programme, Hardware, wie das ROM, das dauerhafte Information enthält, und Firmware, wo die Information relativ dauerhaft ist wie in einem EPROM (siehe dort).
Flip-Flop – eine Schaltung, die einen elektrischen Zustand hält, bis sie durch ein Eingabesignal in den entgegengesetzten Zustand gebracht wird.
Flüchtig – bezieht sich auf Speicher, die ihren Inhalt „vergessen", wenn der Computer abgeschaltet wird.
FORTRAN – Abkürzung für FORmula TRANslation, eine höhere, problemorientierte Programmiersprache für wissenschaftliche und mathematische Zwecke.

G

Gate – Gate = Gatter, eine elektrische Schaltung, die zwar ein oder mehrere Signale empfangen kann, aber nur ein Signal aussendet.
Grafik – Bildinformation im Gegensatz zu Buchstaben und Ziffern.

H

Halbleiter – ein Material, das sonst ein elektrischer Isolator ist, unter bestimmten Umständen aber zum Leiter werden kann.
Hard Copy – Computerausgabe in dauerhafter Form, beispielsweise als ausgedruckte Liste.

Hardware – die physischen Teile des Computers (siehe auch Software und Firmware).
Hexadezimal (Hex) – ein Zahlensystem mit der Basis 16. Die Ziffern Null bis Neun werden ebenso wie die Buchstaben A, B, C, D, E und F dazu verwendet, Zahlen darzustellen. A entspricht 1Ø, B 11, C 12, und so weiter. Hex wird von Mikrocomputer-Anwendern oft verwendet.
Hex Pad – eine Tastatur, die eigens für die Eingabe von Hexadezimalschreibung eingerichtet ist.
höhere Programmiersprache – eine Programmiersprache, die es dem Anwender gestattet, mit dem Computer mehr oder weniger Englisch zu sprechen. Generell kann man sagen: Je höher die Stufe der Sprache (das heißt, je verwandter dem Englischen), desto länger braucht der Computer dazu, sie in eine Sprache umzuwandeln, die er nutzen kann. Sprachen auf niederer Stufe sind für den menschlichen Anwender viel schwieriger, werden in der Regel aber weitaus schneller ausgeführt.

I

Input – die über eine Tastatur, ein Mikrofon, eine Kassette oder eine Diskette in den Computer eingegebene Information.
Input/Output (I/O Device) – Ein-/Ausgabegerät, ein Gerät, das Information oder Befehle von der Außenwelt annimmt, an den Computer weitergibt und nach Verarbeitung die Information in einer Form aussendet, die speicherfähig ist oder von einem Menschen verstanden werden kann.
integrierte Schaltung – eine komplette elektronische Schaltung, eingeätzt auf die Oberfläche eines Halbleiters.

Interface – Interface = Schnittstelle, die Grenze zwischen einem Computer und einem Peripheriegerät, etwa einem Drucker.
Interpreter – ein Programm, das die vom menschlichen Anwender eingegebene höhere Programmiersprache in eine andere Sprache übersetzt, die der Computer verstehen kann.

K

K – bezieht sich auf die Speichergröße. Speicherplatz wird in der Regel in Blöcken von 4K gemessen. 1K enthält 1024 Bytes.
Komparator – eine Schaltung, die zwei Dinge vergleicht und ein Signal liefert, das in Beziehung zu dem Unterschied zwischen den beiden steht.

L

LCD – Abkürzung für Liquid Cristal Diode = Flüssigkeitskristalldiode. Manche Computer, etwa der TRS-80 Taschencomputer haben eine LCD-Anzeige.
LED – Abkürzung für Light Emitting Diode = Leuchtdiode. Die hellen roten Zahlen, oft bei Digitaluhren, bestehen aus LED.
Logik – die mathematische Form einer Untersuchung von Beziehungen zwischen Ereignissen.

M

Maschinensprache oder **Maschinencode** – ein Betriebscode, der vom Computer direkt verstanden und ausgeführt werden kann.

Magnetplatte – siehe Diskette und Floppy Disk.
Mainframe – Computer werden generell in drei Gruppen aufgeteilt. Die Zugehörigkeit hängt mehr oder weniger von der Größe des Computers ab. Der Computer, den Sie sich kaufen wollen, ist ein Mikrocomputer; mittelgroße Computer werden Minicomputer genannt; die riesigen Computer, die Sie manchmal in Science Fiction-Filmen sehen, sind Mainframe-Computer. Bis vor 15 Jahren gab es praktisch nur Mainframe-Anlagen.
Matrixdrucker – ein Drucker, der entweder das Listing eines Programms oder das ausdruckt, was auf dem Bildschirm angezeigt wird. Alle Buchstaben und Symbole bestehen aus einer Anzahl Punkte. Je höher die Zahl der Punkte, desto höher das Auflösungsvermögen des Druckers.
Mehrlagen-Verdrahtungsplatte – Platine, der zusätzliche Schaltungen zugefügt werden können. Diese Schaltungen verleihen dem Computer oft Fähigkeiten, die nicht eingebaut sind, etwa Klangerzeugung oder Steuerung eines Lichtgriffels.
Mikroprozessor – das Herz jedes Computers. Es benötigt periphere Schnittflächen wie Stromzuleitung und Ein- und Ausgabegeräte, damit ein Mikrocomputer zustandekommt.
MODEM – Abkürzung für Modulator Demodulator = Übermittlungseinheit, ein Gerät, durch das zwei Computer über eine Telefonleitung miteinander sprechen können. Verwendet wird meistens ein akustischer Koppler, in den man den Telefonhörer legt.
Monitor – in der Computersprache von zweifacher Bedeutung. Einmal ist ein fernsehartiges Display gemeint. Ein Monitor ist nicht darauf eingerichtet, Fern-

sehprogramme zu empfangen; in der Regel ist das Bild auf einem Monitor besser als das eines normalen Fernsehapparats. Der zweite Sinn eines Monitors betrifft ROM. Als Monitor eines Computers wird die eingebaute Information bezeichnet, wenn Sie ihn erwerben. Mit dieser Information können Entscheidungen getroffen und arithmetische Berechnungen ausgeführt werden.
MPU – Abkürzung für Microprocessor Unit = Mikroprozessor-Einheit.

N

Nanosekunde – eine Nanosekunde ist eine Milliardstelsekunde, Einheit der Geschwindigkeit, nach der ein Computer oder ein Speichertip oft bewertet werden.
Nichtflüchtiger Speicher – Speicher, dessen Inhalt nicht verlorengeht, wenn man den Computer abschaltet. Einige der kleineren Computer, wie der TRS-8Ø Taschencomputer, haben nichtflüchtige Speicher. Die Batterien halten das Programm, das Sie eingeben, mehrere hundert Stunden lang fest.
NOT – eine Boolesche logische Operation, die eine Binärziffer in ihr Gegenteil umwandelt.
Nullstring – auch leerer String. Ein String, der keine Zeichen enthält. In Programmen wird er angezeigt durch zwei doppelte Anführungszeichen, zwischen denen nichts steht.
Numerisch – bezogen auf Zahlen im Gegensatz zu Buchstaben (also alphabetisch). Viele Tastaturen werden als alphanumerisch bezeichnet; sie verfügen über Ziffern und Buchstaben.

O

Oktal – ein Zahlensystem mit 8 als Basis und den Ziffern Ø, 1, 2, 3, 4, 5, 6 und 7. Das Oktalsystem wird heutzutage im Mikrocomputerbereich nicht mehr viel verwendet. Das Hexadezimalsystem ist gängiger (siehe Hexadezimal).

OR – eine arithmetische Operation, die, wenn eine oder mehrere Eingaben 1 sind, 1 liefert.

Oracle – eine Methode, mit einem gesendeten Fernsehsignal Textmitteilungen zu übertragen. Zur Entschlüsselung der Mitteilungen bedarf es einer Teletext-Anlage. Das dialogfähige System ist in Großbritannien entwickelt worden.

Output – Information oder Daten, vom Computer ausgegeben an Geräte wie einen TV-Bildschirm, einen Drucker oder eine Magnetbandkassette.

P

PASCAL – eine höhere Programmiersprache.

Peripherie – alles, was an einen Computer angeschlossen ist, um von ihm gesteuert zu werden, etwa ein Diskettenlaufwerk, ein Drucker oder ein Sprachsynthesizer.

Port – ein Anschluß, durch den Information in den Computer ein- oder von ihm ausgegeben werden kann.

Programm – die Liste der Befehle, die Sie in einen Computer eingeben.

PROM – Abkürzung für Programmable Read Only Memory = programmierbarer Festspeicher. Kann programmiert werden, und das Programm bleibt dort auf Dauer bestehen (siehe auch EPROM und ROM).

R

Random Access Memory (RAM) – RAM = Speicher mit wahlfreiem Zugriff. Der Speicher in einem Computer, der vom Anwender beliebig verändert werden kann. Der Inhalt von RAM geht in der Regel verloren, wenn man den Computer abschaltet. RAM speichert das von Ihnen eingegebene Programm ebenso wie die Ergebnisse von laufenden Berechnungen.
Read-Only Memory (ROM) – ROM = Nur-Lese- oder Festspeicher. Im Gegensatz zum RAM kann Information in ROM vom Anwender nicht verändert werden, und die Information geht nicht verloren, wenn man den Computer abschaltet. Die Daten in ROM werden vom Hersteller eingegeben und teilen dem Computer mit, wie er Entscheidungen treffen und arithmetische Berechnungen ausführen soll. Die Größe von RAM und ROM wird in K angegeben (siehe K).
Rekursion – die ständige Wiederholung eines Programmteils.
Register – ein bestimmter Platz im Speicher, wo während der Operation ein Wort oder mehrere Wörter gespeichert werden.
Reserviertes Wort – ein Wort, das Sie in einem Programm nicht als Variablennamen verwenden können, weil der Computer es als etwas anderes liest. Beispiel: das Wort TO. Weil TO für den Computer einen bestimmten Sinn hat, weisen die meisten Computer es als Variablennamen zurück. Dasselbe gilt für Wörter wie FOR, GOTO und STOP.
Routine – dieses Wort wird gebraucht als Entsprechung für „Programm" oder kann sich auf einen bestimmten

Abschnitt innerhalb eines Programms beziehen (siehe Subroutine).

S

Schleife – eine Folge von Befehlen innerhalb eines Programms, immer wieder ausgeführt, bis eine bestimmte Bedingung erfüllt ist.
Schlüsselwort – das Auslösewort in einer Programmzeile, in der Regel das erste Wort nach der Zeilennummer. Zu den Schlüsselwörtern gehören STOP, PRINT und GOTO.
Seite – oft verwendet als Bezeichnung für die Informationsmenge, die eine Bildschirmseite füllt. Man kann also sagen, man sähe eine Seite aus einem Programm, die Menge des Listings, die gleichzeitig auf dem Bildschirm erscheint.
Seriell – in Folge, Bit für Bit eingespeicherte oder ausgegebene Information.
Signal – ein elektrischer Impuls, der Daten trägt.
Silicon Valley – der gängige Name für ein Gebiet in Kalifornien, wo viele Halbleiter-Firmen ihren Sitz haben.
SNOBOL – eine höhere Programmiersprache.
Software – das Programm, das vom Anwender in den Computer eingegeben wird und ihm mitteilt, was er zu tun hat.
Software-kompatibel – bezieht sich auf zwei verschiedene Computer, die für den jeweils anderen geschriebene Programme verarbeiten können.
Speicher – in einem Computer gibt es zwei Arten von Speichern. ROM (read-only memory = Nur-Lese-Speicher) ist der Speicher, der im Computer bereits pro-

grammiert ist und dem Computer mitteilt, wie er Entscheidungen treffen und arithmetische Berechnungen ausführen soll. Dieser Speicher wird vom Abschalten des Computers nicht betroffen. Der zweite Typ ist RAM (random access memory = Speicher mit wahlfreiem Zugriff). Er nimmt das Programm auf, das Sie über die Tastatur eingeben oder über Kassette oder Diskette überspielen. Bei den meisten Computern „vergißt" der Computer den Inhalt von RAM, wenn Sie abschalten.

Sprache – Computersprachen werden aufgeteilt in drei Bereiche: höhere Programmiersprachen wie BASIC, dem Englischen ziemlich nah und vom Anwender relativ leicht zu gebrauchen, niedere Programmiersprachen wie Assembler, die kurze Bezeichnungen im Zusammenhang mit dem Englischen verwenden (ADD für addieren und RET für Return); und Maschinencode, der mehr oder weniger direkt mit der Maschine verkehrt.

Sprungbefehl – ein Befehl, der den Computer anweist, zu einem anderen Teil des Programms zu gehen, wenn das Ziel dieses Sprungs vom Ergebnis einer eben ausgeführten Berechnung abhängt.

Statistischer Speicher – ein nichtflüssiges Speicheraggregat, das Information behält, solange der Computer läuft, aber keine zusätzlichen Stromstöße benötigt, um den Inhalt zu bewahren.

Subroutine – Subroutine = Unterprogramm, Teil eines Programms, zu dem während der Ausführung des Hauptprogramms oft Zugang erfolgt. Eine Subroutine endet mit einem Befehl, zu der Zeile nach derjenigen zurückzukehren, die das Unterprogramm aufgerufen hat.

T

Teletext – Information, die im oberen Teil eines Fernsehbilds übermittelt wird. Erfordert ein Spezialgerät zur Entschlüsselung, um den Bildschirm mit Text zu füllen.
Terminal – ein Gerät, unabhängig von der Zentraleinheit. Es besteht in der Regel aus einer Tastatur und einer Kathodenstrahlröhre.
Time Sharing – Time sharing = Teilnehmerbetrieb, ein Verfahren, bei dem eine Anzahl von Anwendern Zugang zu einem Großcomputer hat, der in rascher Folge von einem Anwender zum anderen schaltet, so daß jeder Anwender den Eindruck hat, der Computer stehe ihm zu diesem Zeitpunkt allein zur Verfügung.

U

Überlauf – eine Zahl, die für den Computer zu groß oder zu klein ist, als daß der sie verarbeiten könnte.
UHF – Ultra High Frequency (300–3000 Megahertz) = UKW.
Uhr – Zeitgeber im Computer, der seine Operationen synchronisiert.
Ultraviolettlöschung – Zum Löschen von EPROMs (siehe dort) muß UV-Licht verwendet werden.

V

Variable – ein Buchstabe oder eine Kombination von Buchstaben und Symbolen, die der Computer beim Lauf eines Programms einem Wert oder Wort zuteilen kann.
VDU – Abkürzung für Visual Display Unit = Sichtgerät.

W

Wahrheitstabelle – eine mathematische Tabelle, die alle möglichen Ergebnisse einer Booleschen logischen Operation enthält und die Ergebnisse anzeigt, die Sie durch verschiedene Eingabekombinationen erhalten.
Wort – eine Gruppe von Zeichen oder eine Folge von Binärziffern, die eine Informationseinheit darstellen und einen einzigen Speicherplatz besetzen. Der Computer verarbeitet ein Wort als einen einzigen Befehl.
Word-Processor – Textsystem, eine besonders intelligente Schreibmaschine. Der Schreiber kann damit Text verändern, verschieben, Randsteuerung bewerkstelligen und auf dem Bildschirm notfalls ganze Absätze umstellen, bevor die Information an einen Drucker ausgegeben wird. Textsysteme verfügen in der Regel über Speicher, so daß Standardbriefe und die Texte von früher verfaßten Briefen gespeichert werden können.

Z

Zweite Generation – doppelte Bedeutung. Einmal sind Computer gemeint, die Transistoren verwenden, im Gegensatz zu Computern der ersten Generation, die Radioröhren benutzten. Zweite Generation kann auch die zweite Ausgabe eines Programms meinen; nachfolgende Generationen werden durch immer mehr Rauschen geschädigt.



Dieses Buch bietet eine Fülle von nützlichen und wertvollen Anregungen und Programm-Beispielen. Es ist für den Anfänger ebenso wie für den Fortgeschrittenen geeignet.

Das Buch enthält Hinweise zur Verbesserung der Struktur selbstgeschriebener Programme, aber auch »programmierfertige«, universell verwendbare Programme. Dem Leser werden fundierte Informationen über die Arbeitsweise seines ZX81-Computers vermittelt. Ferner gibt der Autor konkrete Hinweise zur Programmierung in Maschinensprache. Hierzu gehören vollständige Programme zum Eingeben in Maschinensprache, Ausgeben von Speicher-Inhalten, Kombinieren von Programm-Modulen.

Der andere Inhaltsschwerpunkt des Buches liegt auf der Beschreibung zahlreicher Spiele, die ausführlich im theoretischen Teil in ihrer Funktion erarbeitet werden.



Hueber Software

ISBN 3-19-008203-0

HUEBER SOFTWARE TASCHENBUCH

Spiele für Ihren ZX 81

Ein Heimcomputer wächst täglich um viele Dimensionen. Bei der Erschließung dieser Dimensionen ist SPIELE FÜR IHREN ZX81 ein hilfreicher Freund. Seine 23 Spiele mobilisieren die Intelligenz und Phantasie des ZX81-Besitzers auf besonders raffinierte und vielfältige Art. Zugleich ist es ein Buch, durch das der Spieler eine Menge dazulernen wird. Durch die Beschäftigung mit den Listings wird er mit vielen Tricks und Techniken vertraut, die dann auch für das individuelle Programmieren nützlich sind. Herausgeber Tim Hartnell und Autor Mark Charlton, zwei der bekanntesten Heimcomputer-Publizisten, halten den Leser ausdrücklich an, die Programme nicht nur in der hier vorgelegten Form durchzuführen. Der wahre Spaß fängt erst damit an, daß man versucht, die Programme nach dem eigenen Geschmack zu variieren und zu verbessern. Es gibt keine »perfekten Programme«, und auch die Spiele dieses Buches können von der Erfindungsgabe und Spielfreude des Benutzers nur profitieren.

Wie die weiteren Bände dieser Reihe — SPIELE FÜR IHREN ZX SPECTRUM, SPIELE FÜR IHREN ATARI, SPIELE FÜR IHREN ORIC, SPIELE FÜR IHREN DRAGON, SPIELE FÜR IHREN BBC-MICRO, SPIELE FÜR IHREN VIC 20 und SPIELE FÜR IHREN TRS — enthält SPIELE FÜR IHREN ZX81 ein kleines Lexikon mit der Erklärung der wichtigsten Computer-Begriffe.

ISBN 3-19-008212-X 8212 DM 14.80

Hueber Software Taschenbuch

Spiele für Ihren ZX 81

AMSTRAD CPC

MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA



https://acpc.me/